# A TIME ACCURATE FLUID-STRUCTURE INTERACTION FRAMEWORK USING A CARTESIAN GRID CFD SOLVER

A Thesis
Presented to
The Academic Faculty

by

Matthew Scott Bopp

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Daniel Guggenheim School of Aerospace Engineering

Georgia Institute of Technology
December 2017

# A TIME ACCURATE FLUID-STRUCTURE INTERACTION FRAMEWORK USING A CARTESIAN GRID CFD SOLVER

Approved by:


Dr. Stephen M. Ruffin, Advisor
Daniel Guggenheim School of Aerospace Engineering
*Georgia Institute of Technology*

Dr. Lakshmi N. Sankar
Daniel Guggenheim School of Aerospace Engineering
*Georgia Institute of Technology*

Dr. Michael D. Barnhardt
Aerothermodynamics Branch
*NASA Ames Research Center*

Dr. Marilyn J. Smith
Daniel Guggenheim School of Aerospace Engineering
*Georgia Institute of Technology*

Dr. Julián J. Rimoli
Daniel Guggenheim School of Aerospace Engineering
*Georgia Institute of Technology*



Date Approved: 9 November 2017

# ACKNOWLEDGEMENTS

This thesis represents years of work, and would not be possible without the support I've received from everyone along the way. Though I cannot personally mention everyone by name, I am truly thankful for each and every one of you. First, I would like to thank my advisor, Dr. Stephen Ruffin, for his continued support and encouragement over the years. He provided guidance when sought, and the freedom to set my own goals and work through the challenges, allowing me to develop the necessary skills required to successfully perform independent research. Dr. Michael Barnhardt, who I can now call a friend, was a source of mentorship during my first internships at NASA Ames, which led to several years of collaboration through the NASA Space Technology Research Fellowship program. My remaining committee members, including Dr. Lakshmi Sankar, Dr. Marilyn Smith, and Dr. Julián J. Rimoli, provided valuable feedback throughout this process. I would like to thank them for their advice and insight. Dr. David Hash, who provided me with my first opportunity to spend a summer at NASA Ames, was the catalyst for the relationship I developed with NASA. His support allowed me to grow as a researcher, and I thank him for the many philosophical discussions we had. I can also give him credit for when I finally caved in, and found myself drinking a cup of coffee from time to time, as I pushed towards completion of this work.

This research was made possible primarily under the support of a NASA Space Technology Research Fellowship. I would like to thank the NASA Office of the Chief Technologist for providing the opportunity to explore this research topic, and for providing the means for working alongside NASA researchers at both NASA Ames and NASA Langley. While at NASA Ames, I also worked with Dr. Eric Stern, who I would like to thank for his collaboration efforts. I would like to express gratitude to Dr. Sasan Armand, who provided guidance while I spent portions of two summers at NASA Langley. The fellowship program also put me in connection with fellow students from a wide range of academic institutions,

for whom I am thankful to have met. Additionally, I would like to acknowledge the Georgia Space Grant Consortium for their support of this research as well.

I would like to thank all of my lab mates in the Aerothermodynamics Research and Technology Lab (ARTLab). There are too many to list, but I am sincerely grateful to you all for the friendship and academic discussions. Dr. Susheel Sekhar and Alex Pace provided the necessary guidance as the senior lab members when I began my graduate studies. I must thank Chandan Prasad for his initial role in the re-development of NASCART-GT, which further sparked my interests in computer science and in collaborative, computational tool development. David Dement was a great source for bouncing ideas off of, especially anything related to numerical algorithms.

The support from many friends outside of this academic pursuit cannot go unmentioned. These include lifelong friends, through the most recent friendships I've developed in Atlanta. I still get a kick out of their image of me working in "the lab", wearing a white lab coat, and using a Bunsen burner. I should have just referred to it as "the office". I'm especially thankful for the relationships I've developed within the cycling community of Atlanta. They provided distractions when necessary, but more importantly, encouragement and support, especially near the completion of my graduate studies.

Finally, but most important, I would not be where I am without the love and support from my family. My parents, Phyllis and Jim Bopp, have always been encouraging, and I thank them for everything. My brother, Chris, has been a true friend, and I'd like to thank him for the many philosophical discussions through the years.

# TABLE OF CONTENTS

**ACKNOWLEDGEMENTS** . . . . . **iii**

**LIST OF TABLES** . . . . . **viii**

**LIST OF FIGURES** . . . . . **ix**

**SUMMARY** . . . . . **xvi**

**I INTRODUCTION** . . . . . **1**

1.1 Background and Motivation . . . . . 1

1.2 Computational Overview . . . . . 4

1.3 Goal & Objectives . . . . . 19

**II GOVERNING EQUATIONS AND NUMERICAL IMPLEMENTATION 21**

2.1 Governing Equations for Fluid-Structure Interactions . . . . . 21

2.2 Geometric Conservation Law . . . . . 22

2.3 Fluid Dynamics . . . . . 23

2.4 Structural Dynamics . . . . . 26

2.5 Fluid-Structure Coupling Methods . . . . . 26

2.6 Fluid-Structure Interface . . . . . 32

2.6.1 CFD Data Interpolation to Input Geometry . . . . . 32

2.6.2 Similar Meshes at the Interface . . . . . 33

2.6.3 Dissimilar Meshes at the Interface . . . . . 34

2.6.4 Discrete Data Transfer Between Dissimilar Meshes (DDTBDM) . . 35

2.7 CFD Solver . . . . . 36

2.7.1 Motivation for New Software Development . . . . . 37

2.7.2 Code Design . . . . . 39

2.7.3 High Level Structure . . . . . 40

2.7.4 Geometry Configuration and Motion . . . . . 41

2.7.5 Grid Generation . . . . . 43

2.7.6 Time Integration . . . . . 45

2.7.7 Spatial Scheme . . . . . 45

2.7.8 Convective Flux Schemes . . . . . 47

2.7.9 Viscous Flux Scheme . . . 48
2.7.10 Source Terms . . . 48
2.7.11 Boundary Conditions . . . 48
2.7.12 Solution-Based Grid Adaption . . . 53
2.7.13 Variable and Mixture Classes . . . 54
2.8 FEA Solver . . . 54
**III STATIONARY & PRESCRIBED MOTION SIMULATIONS . . . 56**
3.1 Rigid Tension Cone . . . 56
3.1.1 Rigid Tension Cone: $0^o$ Angle of Attack . . . 57
3.1.2 Rigid Tension Cone: $20^o$ Angle of Attack . . . 59
3.2 Constant Velocity Piston . . . 60
3.2.1 Fine Mesh Results . . . 61
3.2.2 Discussion on Accuracy . . . 67
3.3 Supersonic Diamond Airfoil . . . 78
3.4 Supersonic Cylinder . . . 81
3.5 Transonic Oscillating NACA 0012 Airfoil . . . 84
**IV FREE FLIGHT SIMULATIONS . . . 89**
4.1 Physical Problem and Computational Mesh . . . 90
4.2 Static Analysis Verifications . . . 92
4.3 Free Flight Analysis . . . 98
**V FLUID-STRUCTURE INTERACTIONS . . . 102**
5.1 Dynamic Analysis of Elastic Piston . . . 102
5.1.1 Elastic Piston with 0-D Fluid Model . . . 103
5.1.2 Elastic Piston with 1-D Fluid Model . . . 109
5.2 Vortex-Induced Vibrations of an Elastic Cylinder . . . 113
5.2.1 Static Cylinder . . . 114
5.2.2 Elastic Cylinder . . . 115
5.3 Shock Wave Impacting a Deforming Panel . . . 117
5.3.1 Fluid Dynamic Analysis of Rigid Configuration . . . 118
5.3.2 Structural Dynamics . . . 123

5.3.3 Coupled Analysis of Deforming Panel . . . . . 124
5.4 Steady-State FSI Analysis of Semi-Rigid Tension Cone . . . . . 128
5.4.1 Structural Analysis . . . . . 129
5.4.2 Fluid Dynamic Analysis . . . . . 132
5.4.3 Mesh Mapping and Data Transfer . . . . . 136
5.4.4 FSI Simulations . . . . . 141
5.5 Time Accurate Analysis of Semi-Rigid Tension Cone . . . . . 150
5.5.1 FEA Analysis . . . . . 150
5.5.2 CFD Analysis . . . . . 151
5.5.3 FSI Analysis . . . . . 152
**VI CONCLUSIONS AND FUTURE WORK . . . . . 158**
6.1 Overview . . . . . 158
6.2 Results & Conclusions . . . . . 159
6.3 Suggestions for Future Work . . . . . 161
**APPENDIX A — ALE FORMULATION OF AUSMPW+ . . . . . 164**
**APPENDIX B — 6-DOF MODEL . . . . . 167**
**APPENDIX C — HARMONIC OSCILLATOR IMPLEMENTATION . 169**
**APPENDIX D — NEWMARK INTEGRATION SCHEME . . . . . 175**
**APPENDIX E — ENTROPY WAVE ADVECTION . . . . . 176**
**REFERENCES . . . . . 180**

# LIST OF TABLES


1 Riemman invariants for a perfect gas. . . . . 52

2 Geometry and flow conditions of AGARD CT Case 5. . . . . 86

3 Comparison of pitch frequencies. . . . . 100

4 Comparison of pitch frequencies for grid convergence study. . . . . 101

5 Material properties for the semi-rigid tension cone. . . . . 131

6 Flow conditions for the semi-rigid tension cone simulation. . . . . 135

## LIST OF FIGURES


1 Parachute drag coefficient variation with Mach number (Ref. [107]). . . . . 2

2 Examples of aerodynamic decelerator configurations [101]. . . . . 4

3 IAD configuration for FEA and CFD for a contiuum simulation [93]. The structural dynamics solution computed using LS-DYNA is shown in (a), while the flowfield solution computed using NASCART-GT is shown in (b). . . . 6

4 Example flow solution presented by Lee [68], in which chemical nonequilibrium has been accounted for. $N_2$ mole fractions are shown on x-plane and N mole fractions on y-plane. . . . . 7

5 Disk-Gap-Band parachute configuration used in the LES FSI analysis [56]. . 9

6 Partitioned coupling strategy, where the fluid dynamics (CFD) and structural dynamics (FEA) are computed using separate solvers. . . . . 28

7 Diagrams depicting the Conventional Serial Staggered (CSS) method shown in (a) and CSS method using subcycling in (b). . . . . 29

8 Diagrams depicting two partitioned coupling schemes, with the GSS method 1 shown in (a) and the GSS method 2 shown in (b). . . . . 32

9 Comparison between an FEA surface mesh and a CFD surface mesh, taken as a slice along the symmetry plane in order to highlight the topological differences. This is an example of a tension cone configuration. . . . . 35

10 Cell-octree data structure. . . . . 43

11 Inheritance diagram of the Scheme class. . . . . 46

12 Stencil construction for (a) structured approach and (b) unstructured approach for extrapolating the state to the faces of cell # 0. . . . . 48

13 Generic numerical stencil used near surface. . . . . 50

14 Comparison between the computed density field and the experimental schlieren image for the rigid tension cone at $0^o$ angle of attack. . . . . 58

15 Pressure coefficient along the symmetry plane of the rigid tension cone at $0^o$ angle of attack. . . . . 58

16 Comparison between the computed density field and the experimental schlieren images for the rigid tension cone at $20^o$ angle of attack. . . . . 59

17 Pressure coefficient along the symmetry plane of the rigid tension cone at $20^o$ angle of attack. . . . . 60

18 Density distribution for stationary, 1-D piston simulation, where NASCART-GT result is compared against analytic solution. . . . . 62

19 Detailed view of the density distribution in the region surrounding either side of the stationary piston. . . . . 63

20 Velocity distribution for stationary, 1-D piston simulation, where NASCART-GT result is compared against analytic solution. . . . . 63

21 Pressure distribution for stationary, 1-D piston simulation, where NASCART-GT result is compared against analytic solution. . . . . 64

22 Density distribution for translating, 1-D piston simulation, where NASCART-GT result is compared against analytic solution. . . . . 65

23 Detailed view of the density distribution in the region surrounding either side of the piston. . . . . 65

24 Velocity distribution for translating, 1-D piston simulation, where NASCART-GT result is compared against analytic solution. . . . . 66

25 Pressure distribution for translating, 1-D piston simulation, where NASCART-GT result is compared against analytic solution. . . . . 66

26 Pressure distribution through the shock wave and expansion, as a function of mesh refinement level (Ref. Level) for the stationary piston. . . . . 68

27 Pressure distribution through the shock wave and expansion as a function of mesh refinement level (Ref. Level) for the translating piston. . . . . 68

28 Density distribution through shock wave and expansion for mesh refinement level 7. Comparisons are made between $1^{st}$ and $2^{nd}$ order spatial accuracy for the stationary and translating piston. . . . . 69

29 Pressure distribution through shock wave and expansion for mesh refinement level 7. Comparisons are made between $1^{st}$ and $2^{nd}$ order spatial accuracy for the stationary and translating piston. . . . . 70

30 $L_1$ and $L_2$ error in density as a function of cell size for the compression side of the flowfield. . . . . 74

31 $L_1$ and $L_2$ error in density as a function of cell size for the expansion side of the flowfield. . . . . 74

32 Comparison between density and pressure distributions through the shock wave as a function of piston velocity, while relative velocity between piston and fluid remains constant. . . . . 75

33 Comparison between density and pressure distributions through the expansion wave as a function of piston velocity, while relative velocity between piston and fluid remains constant. . . . . 76

34 Velocity distribution in the region between piston and expansion waves. Axial position for each simulation is shifted such that piston location matches the stationary piston simulation. The velocity label along the right corresponds to piston speed, and the dashed line indicates the value of the local speed of sound at the foot of the expansion. Symbols represent mesh nodes. . . . . 77

35 $L_1$ error in density as a function of piston velocity, while relative velocity between piston and fluid remains constant. Results are shown for both the shock and expansion regions for two mesh refinement levels. . . . . 78

36 Comparison between a stationary and translating diamond airfoil on a uniform mesh. The pressure flowfield is shown in (a) and the surface pressure distribution is shown in (b). . . . . 80

37 Comparison between a stationary and translating diamond airfoil on a non-uniform mesh. The pressure flowfield is shown in (a), with the addition of mesh lines shown in (b). . . . . 81

38 Comparison of surface pressure distribution between a stationary and translating diamond airfoil for a non-uniform mesh (a) and a comparison between all diamond airfoil results (b). . . . . 81

39 Comparison between a stationary and translating supersonic cylinder on a uniform mesh. The pressure flowfield is shown in (a) and the surface pressure distribution is shown in (b). . . . . 82

40 Comparison between a stationary and translating supersonic cylinder on a non-uniform mesh. The pressure flowfield solution is shown in (a), along with the grid lines in (b). . . . . 84

41 A comparison of surface pressure distribution between a stationary and translating supersonic cylinder on a non-uniform mesh is shown (a). Stagnation line pressure profiles are shown for all supersonic cylinder simulations in (b). 84

42 Snapshots of flowfield pressure distributions for the relative motion, oscillating NACA 0012 airfoil. . . . . 87

43 Normal force coefficient as a function of angle of attack for an oscillating NACA 0012 airfoil. Results are compared against experimental data from Ref. [65]. . . . . 88

44 Comparison of meshes between NASCART-GT and US3D. . . . . 90

45 Comparison of meshes between NASCART-GT and US3D. The mesh in (a) highlights the differences within the shock layer, and the mesh in (b) compares the cell sizes near the surface of the capsule and in the region of the bow shock. . . . . 91

46 Aerodynamic load comparisons between NASCART-GT and US3D for the 2-D MSL verification simulations. The force coefficient is shown in (a) and the absolute force coefficient error is shown in (b). . . . . 93

47 Flowfield comparison between NASCART-GT and US3D for $\alpha = 0^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours. . . . . 93

48 Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 0^o$ simulation. . . . . 94

49 Flowfield comparison between NASCART-GT and US3D for $\alpha = 10^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours. . . . . 94

50 Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 10^o$ simulation. . . . . 95

51 Flowfield comparison between NASCART-GT and US3D for $\alpha = 20^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours. . . . . 95

52 Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 20^o$ simulation. . . . . 96

53 Flowfield comparison between NASCART-GT and US3D for $\alpha = 30^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours. . . . . 96

54 Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 30^o$ simulation. . . . . 97

55 Comparison of meshing strategies of NASCART-GT against US3D simulation. 99

56 Grid convergence study. . . . . 100

57 Grid converged simulation, showing the pitch angle history, as well as the time history of the force coefficients in the x and y directions. . . . . 101

58 Model diagram of 1-D elastic piston configuration. . . . . 103

59 Displacement and pressure histories for the simulation of the 0-D linear piston. 106

60 $L_1$ displacement error for the simulations of the linear forcing function. . . 107

61 Displacement and pressure histories for the simulation of the 0-D piston using the linear and nonlinear forcing functions. . . . . 108

62 $L_1$ displacement error for the simulations of the non-linear forcing function. 109

63 Displacement history for the 1-D elastic piston. The ISS1 coupling scheme with $\Delta t = 2.5 \times 10^{-6} s$ is used for both simulations. Solutions are compared against reference computational data [9] and against analytic eigenfrequency 111

64 $L_1$ displacement error for the 1-D elastic piston. . . . . 112

65 Diagram depicting cylinder of mass, $m$, attached to springs in each coordinate direction. The spring stiffness, $k$, and the damping constant, $c$, are the same for both springs. . . . . 113

66 Flowfield for static cylinder, highlighting the vortex pattern in the wake. . . 115

67 Time history of lift coefficient for the static cylinder simulation. . . . . 115

68 Displacement time histories and phase plots for elastic cylinder simulation. Comparison made against reference simulation [118]. . . . . 117

69 Experimental schematic (Giordano et al. [44]). . . . . 117

70 Initial conditions for the left and right side of discontinuity. Flowfield boundary conditions are also listed. Note that the left side of the domain actually extends to an x-value of -0.505 $m$, but not shown here for the sake of clarity. 118

71 Rigid panel flowfield showing density contours. Time intervals selected to show key flow features. Solution was computed using the fine uniform mesh. 119

72 Time history of computational surface loads for three grid refinement levels. 121

73 Time history of pressure for three grid refinement levels. . . . . 121

74 Three different initial mesh strategies were considered in the CFD analysis. (a) utilizes the typical meshing strategy with refinement near the surface, and is treated as the baseline case. (b) adds a region of uniformly refined cells in the location of the pressure sensor. (c) is a uniformly refined mesh in the region containing the base and panel. . . . . 122

75 Comparison of computational surface loads using fine mesh for three refinement strategies. . . . . 123

76 Comparison of pressure history at sensor location using fine mesh for three refinement strategies. . . . . 123

77 Time history comparison of pressure at the sensor for the 40 mm panel. Comparisons are made against the experimental and numerical results of Giordano et al. [44]. . . . . 125

78 Comparison of present results with the experimental and numerical results of Giordano et al. [44] and Brehm et al. [12] . . . . . 126

79 Comparison of time history of pressure at sensor for 50 mm panel against the experimental and numerical results of Giordano et al. [44] . . . . . 127

80 Comparison of present results for 50 mm panel with the experimental and numerical results of Giordano et al. [44] and Brehm et al. [12] . . . . . 128

81 Images of the experimental semi-rigid tension cone, as shown in the reference work by Clark [22]. . . . . 129

82 The LS-DYNA FEA mesh is shown with parts indicated by different colors. The tension shell is broken into two parts: the primary shell (red) and the seams (blue). The torus is green and the sting connector is yellow. . . . . 130

83 Steady state solutions with an applied aerodynamic load. . . . . 132

84 CFD input geometry configuration consists of six components: 1. aeroshell (gray), 2. tension shell (red), 3. torus (green), 4. rear tension shell (blue), 5. sting connector (yellow), 6. sting (orange). . . . . 133

85 CFD input geometry as viewed from the negative Z side of the symmetry plane. . . . . 134

86 The initial mesh around the CFD model is shown with (a) depicting the 3-D model and (b) highlighting the mesh near the surface along the Z-symmetry plane. . . . . 134

87 Steady state CFD solution. The flowfield is colored by Mach number, while the surface pressure coefficient is shown on the surface. . . . . 136

88 Front view of CFD and FEA meshes with part identification numbers used for data mapping. . . . . 138

89 Back view of CFD and FEA meshes with part identification numbers used for data mapping. . . . . 138

90 Aerodynamic loads are shown in the form of surface pressure coefficient, with the corresponding nodal force vectors on the FEA mesh. . . . . 139

91 Displacement distributions are shown on the front side of the FEA and CFD meshes. . . . . 140

92 Displacement distributions are shown on the back side of the FEA and CFD meshes. . . . . 140

93 Flowfield slice along symmetry plane showing final density solution for the $0^o$ angle of attack. . . . . 142

94 Comparison of symmetry plane deformation between initial geometry, FSI converged geometry, and experimental data. . . . . 143

95 CFD normalized RMS residual history. . . . . 144

96 CFD axial load history computed over entire input geometry configuration. 144

97 Axial displacement history of the torus. . . . . 145

98 Flowfield slice along symmetry plane showing final density solution for the $9^o$ angle of attack. . . . . 146

99 Comparison of symmetry plane deformation between initial geometry and FSI converged geometry for the $9^o$ angle of attack simulation. . . . . 147

100 Close-up views of the upper and lower portions of the tension cone for the $9^o$ angle of attack simulation. . . . . 147

101 CFD normalized RMS residual history for the $9^o$ angle of attack simulation. 148

102 CFD axial load history computed over entire input geometry configuration for the $9^o$ angle of attack simulation. . . . . 149

103 Axial displacement history of the torus for the $9^o$ angle of attack simulation. 149

104 CFD input geometry configuration consists of six components: 1. aeroshell (gray), 2. tension shell (red), 3. torus (green), 4. rear tension shell (blue), 5. backing plate (yellow). . . . . 151

105 CFD input geometry as viewed from the opposite side of the symmetry plane. These views illustrate how the tension cone is constructed for the time accurate simulations. . . . . 152

106 Flow solution of nominal dynamic FSI simulation. The flowfield is colored by Mach number and the surface by pressure coefficient. . . . . 153

107 Axial force coefficient of the entire configuration plotted against simulation time. The effects of material thickness are shown in (a), while the effects of freestream Mach number are shown in (b). . . . . 154

108 Comparison of nodal displacement in the axial direction as a function of material thickness. Node 62 is shown in (a) and node 224 in (b). . . . . 155

109 Comparison of nodal displacement in the axial direction as a function of freestream Mach number. Node 62 is shown in (a) and node 224 in (b). . . 156

110 Comparison of Von-Mises stress as a function of material thickness. Element 2886 is shown in (a) and element 3126 in (b). . . . . 156

111 Comparison of Von-Mises stress as a function of freestream Mach number. Element 2886 is shown in (a) and element 3126 in (b). . . . . 157

112 The displacement history is shown as a result of the numerical integration of the undamped, free response, mass-spring system. . . . . 170

113 Time history of $L_1$ error. The displacement history is added to provide a reference. The $L_1$ error is shown for three different time steps, with the maximum error within each period of oscillation included. As the simulation progresses in time, the maximum displacement maintains a high level of accuracy compared to the points in time when the displacement is near zero, and the mass is moving at maximal velocity. . . . . 171

114 Time step convergence shown for three different time steps, processed at three separate instances in time. . . . . 171

115 The displacement history is shown as a result of the numerical integration of the forced mass-spring system. . . . . 173

116 Time history of $L_1$ error. The displacement history is added to provide a reference. The $L_1$ error is shown for three different time steps, with the maximum error within each period of oscillation included. . . . . 173

117 Time step convergence shown for three different time steps, processed at three separate instances in time. . . . . 174

118 Entropy wave advection: initial condition. . . . . 177

119 $L_1$ error for entropy wave advection simulations. . . . . 178

120 $L_2$ error for entropy wave advection simulations. . . . . 178

121 $L_\infty$ error for entropy wave advection simulations. . . . . 179

# SUMMARY

The landing of the Mars Science Laboratory (MSL) in 2012 demonstrated the limits of supersonic planetary entry technology through the use of a disk-gap-band parachute deployed from behind the aeroshell capsule. With the eventual goal of sending humans to Mars, the payload requirements are estimated to increase by a factor of 40, far outside the current technological envelope. With a density of less than 1% of Earth's, the Martian atmosphere makes the task of generating aerodynamic drag very challenging. Larger aeroshells produce more drag, but the vehicle is then too large to fit as payload inside a rocket. By utilizing inflatable aerodynamic decelerators, the drag area can be significantly increased, while the pre-deployed configuration has high packing efficiency.

New technologies bring with them the requirement to study their behavior, and characterize their flight limits. Wind tunnel tests are difficult due to scaling concerns, and flight tests are costly and time consuming. Thus, accurate computational modeling of the fluid-structure interactions (FSI) is critical in the development of aerodynamic decelerators. Much of the current research in FSI focuses on high fidelity analysis, which is often very computationally expensive, and requires significant user intervention. The current work fills a niche where the analysis time and human interaction is reduced, by utilizing an adaptive, Cartesian grid framework for solving the computational fluid dynamics (CFD).

A time accurate, partitioned coupling strategy is employed to study FSI applied to flexible materials under high dynamic pressure loads. The structural dynamics is solved using LS-DYNA, and care must be taken at the interface boundary conditions to reduce numerical errors. The development of this tool has relied on a complete re-write of the in-house CFD code, NASCART-GT, where significant improvements have been made in computational performance and efficiency.

The first set of results explore the numerical accuracy of stationary and prescribed motion CFD simulations. Supersonic flow over a rigid tension cone demonstrates the accuracy of the inviscid solver to predict the shock structure and the surface pressure distribution over a flight relevant configuration. The effects of non-stationary geometries are studied by examining several configurations. Spatial order of accuracy is explored for the 1-D translating piston. It is found that in order to maintain similar accuracy to stationary input geometries, a finer mesh resolution is required for moving geometries to capture the propagation of shock waves. An important conclusion that is drawn is that the numerical accuracy decreases as the geometry velocity increases in regions of flow compression, while the numerical accuracy increases as the geometry velocity increases in regions of flow expansion. Supersonic flow past a translating diamond airfoil and a translating cylinder both demonstrate the necessity for robust solution adaption techniques in order to accurately predict multi-dimensional flows. The prescribed pitching motion of a NACA 0012 airfoil in transonic flow validates the capability of the methodology in computing transient aerodynamic loads.

The next step towards FSI problems examines the rigid body dynamics of a 2-D MSL capsule. Static aerodynamics are validated against solutions from a reference numerical solver. Analysis of the free flight motion highlights the importance of solution adaption and mesh refinement in predicting the time history of the pitch angle. The pitching frequencies show convergence with a mesh refinement study, and the negligible pitch damping demonstrates the low numerical dissipation that is produced during these simulations.

Time accurate FSI simulations are studied by investigating several physical problems. The coupling between a simplified 0-D fluid model and an elastic piston provides a means for validating the numerical accuracy for several coupling strategies up to $2^{nd}$ order accuracy. The 1-D problem is then presented, which yields at most $1^{st}$ order accuracy, as a result of the $1^{st}$ order accurate Euler simulations. Next, shock wave interactions with a deforming panel are simulated, and results demonstrate that the coupling methodology is capable of accurately predicting the time history of the flowfield pressure and the displacement of the upper end of the panel.

The final set of simulations first validate the steady-state FSI of a semi-rigid tension cone against experimental data. The time accurate, dynamic response of this 3-D problem is then presented, where notional flight conditions are used. The dynamic response for these simulations is characterized by studying the displacement and stress histories, as well as the aerodynamic loads over time. The effects of Mach number and material thickness are explored, demonstrating that the present methodology can serve as a predictive computational tool for the time accurate FSI analysis of inflatable decelerators.

# CHAPTER I

# INTRODUCTION

## *1.1 Background and Motivation*

During the entry, descent, and landing (EDL) phase of a planetary exploration mission, the process of deceleration and selection of an aerodynamic decelerator is critical. An aerodynamic decelerator can be any device used to reduce the ballistic coefficient of a spacecraft, by leveraging the aerodynamic effects of drag. The flight conditions where such devices are necessary range from transonic to hypersonic Mach numbers. At these conditions, the device must be designed not only to aerodynamic specifications, but also to potentially withstand a high heating environment produced in the hypersonic regime. There are several important goals, which aerodynamic decelerators must be designed to achieve [26]: (1) the deceleration from supersonic to subsonic speeds, (2) minimization of descent rate in order to reduce loading on the vehicle, (3) enhance stability, which maintains proper vehicle orientation, (4) work efficiently with additional aerodynamic deployment devices, (5) reduce the ballistic coefficient, and (6) increase altitude and timeline in order to allow for more time to complete the EDL sequence of events.

Various deceleration approaches have been studied, tested, and flown over the past 50 years. The parachute, which has been used for all U.S. planetary entries, is designed to be deployed at Mach numbers at or below two, depending on the dynamic pressure. Higher Mach numbers and dynamic pressures, however, have been known to cause greater instability during parachute inflation [42]. In addition, the drag coefficient of the parachute decreases with increasing Mach number. Other key parameters in designing a parachute include the fabric porosity [77] and towing distance [19, 76].

In addition to the aforementioned limitations, parachutes also exhibit an undesirably significant decrease in drag coefficient in the transonic regime. The characteristic drag coefficient profile as a function of Mach number is well represented by the plot in Fig. 1,

which has been re-produced from Refs. [4] and [107]. The drag coefficient is approximately constant in the subsonic regime, falls significantly near Mach 1, before increasing to a local maximum at low supersonic Mach numbers.

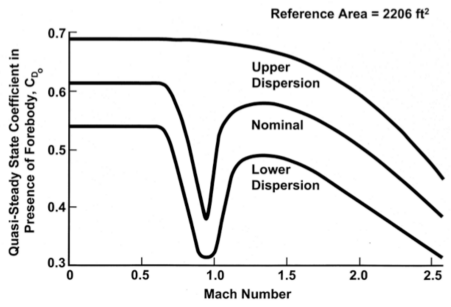



Figure 1: Parachute drag coefficient variation with Mach number (Ref. [107]).

All planetary entry missions up to the year 1990 did not experience flight conditions outside of the effective range of applicability for parachutes. These missions included Viking, Pioneer Venus, and Galilieo. Since 1990, missions have entered the Martian atmosphere, Titan atmosphere, and Earth's atmosphere (sample return missions), using disk-gap-band (DGB) parachutes as their primary deceleration device.

Improvements to parachute technology have been achieved incrementally, over the past 30 years. The limits of this technology have been rapidly approached, as missions require significant increases in landed mass [11]. The Mars Science Laboratory (MSL) was launched in 2011, and passed through the Martian atmosphere on its way to the surface in August 2012. In order to land the nearly 900 kg payload safely on the Martian surface, the largest aeroshell and parachute ever flown was used. With parachute deployment at a higher Mach number and dynamic pressure than previously flown [23], current and future missions must perform in uncharted territory.

With the desire to land even larger payloads on Mars and other planets, significant research and development has been necessary. The two main approaches include deployable decelerators and supersonic retropropulsion (SRP). The latter approach requires the installation of propulsion devices oriented to produce thrust in the opposite direction of the vehicle's motion. The approach driven by deployable decelerators has also been a heavily

researched concept, and is the motivating configuration for the present work. Specifically, the technology is referred to as an inflatable aerodynamic decelerator (IAD). An IAD can be defined as an inflatable device designed to greatly increase drag on an entry vehicle [101]. IADs are classified as being Supersonic IADs (SIAD) or Hypersonic IADs (HIAD). The former is designed for Mach numbers that fall within the supersonic regime, and must be able to withstand only minimal heating and moderate to high dynamic pressures. HIADs are designed to be deployed prior to atmospheric entry, and flown within the hypersonic Mach regime. The key difference in the latter configuration is that it must be able to withstand a much higher dynamic pressure and thermal loading.

There are various IAD configurations that have been studied and tested over the years, each with its own design advantages. IADs are generally broken into two categories: trailing IADs and attached IADs. Figure 2 shows examples of each kind. The primary attached IAD configurations are the isotensoid, tension cone, and stacked toroid. The isotensoid has derived its name from its structural characteristics in that all the fabric stress is uniform in all directions. The shape was derived by Houtz [51] for a given pressure distribution. Thus, the shape is designed based on notional flight conditions. The tension cone is based on a similar concept in that the applied pressure distribution determines the shape. The shape was derived based on an assumed axisymmetric pressure distribution by Anderson [2]. The underlying intention of this shape is that the shell remains under tension at all times, assuming that the ratio of circumferential to meridional stress is constant. The advantage of this is that the structure tends to resist deformations, which has the effect of simplifying the extent of fluid-structure coupling that is required. One end of the tension cone is attached to the vehicle while the aft end is attached to a either a rigid or inflated compression ring. The stacked toroid consists of a series of inflated torii that are layered in increasing size. A thermal protection system coats the exterior so the forebody becomes a smooth, conical shape.

Trailing IADs are configurations in which an inflatable device is towed at some distance behind the primary entry vehicle. Examples of such configurations are also illustrated in Fig. 2. The shapes can be as simple as a sphere, or be of similar shapes to their attached

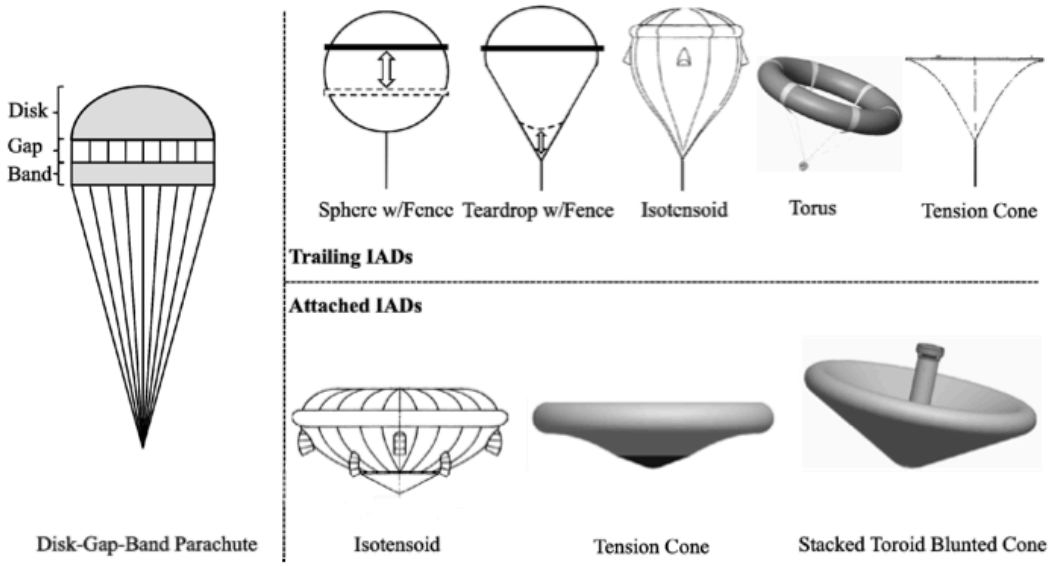



Figure 2: Examples of aerodynamic decelerator configurations [101].

counterparts. The prediction of trailing decelerator dynamics can often be more complex since the IAD is in the presence of the vehicle's wake, which may be an unsteady, turbulent region. Accurate wake flow calculations can come at the cost of expensive computational effort, but are required to accurately predict the drag performance. Aerodynamic stability requires extra attention due to the balance of forces between the two-body configuration. Generally, trailing IADs generate less drag due to the lower pressure within the wake flow [101]. Smith, et al. [101] provides a comprehensive discussion of the different configurations and their historical development.

With the need for detailed understanding of new deceleration systems, increased efforts in experimental and computational analyses are required. The next section describes the recent work that has been undertaken in an effort towards advancing deceleration technology. This perspective lays the groundwork for the needs and requirements to further advance this area of research.

### *1.2 Computational Overview*

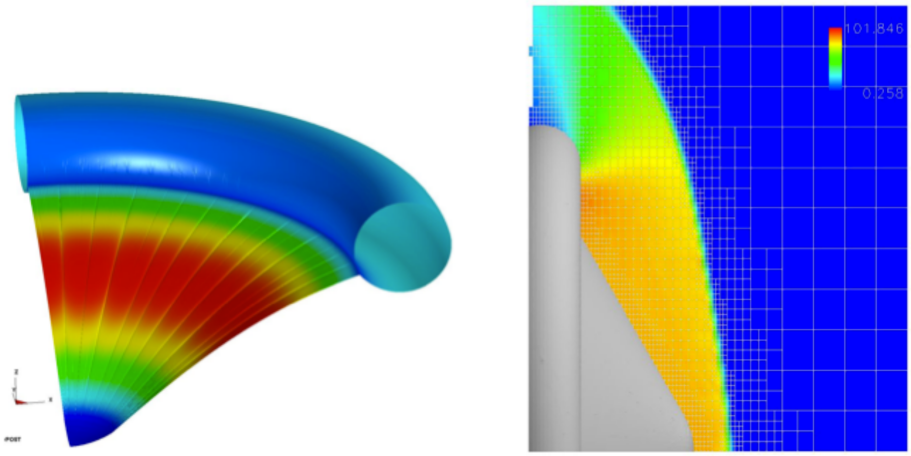
Over the past couple decades, various efforts have been focused on the advancement of computational analyses, experimental testing, and flight testing. Computational analysis has increasingly become more capable in simulating both the complex flow environments,

as well as the ability to couple multiple disciplines to provide more accurate simulations. Experimental testing has predominantly consisted of static measurements, but with the increased need to validate unsteady FSI, time-varying measurements are desirable.

An overview of the current state of aerodynamic decelerator FSI was presented at the 2011 AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar [91]. The AIAA-ADS (American Institute of Aeronautics and Astronautics - Aerodynamic Decelerator Systems) sub-committee on Advanced Modeling Development and Validation prepared a white paper exploring the current computational modeling capabilities of FSI tools for applications such as parachutes and inflatable decelerators. Details regarding the numerical requirements for running such simulations were discussed, and current challenges were highlighted. One of the most important requirements is for the development of experimental databases that can lead to proper verification and validation of numerical models. Several challenges were highlighted including the highly nonlinear nature of large structural deformations, largely varying Mach regimes, and research overhead. The continued research provided by the present work also provides a significant contribution to the body of FSI work, that the sub-committee has been seeking.

Rohrschneider [93] developed a multi-fidelity analysis tool to investigate the aeroelastic behavior in the continuum, transitional, and rarefied flow regimes. The low fidelity aerodynamic analysis used the developed tool called the Moving Boundary Simple Aerodynamics (MBSA) code. The code combined the modified Newtonian approach in the continuum regime with the collisionless DSMC method developed by Bird [8]. The high fidelity aerodynamics in this analysis refers to the inviscid perfect gas computation using NASCART-GT in the continuum regime and DAC [66] in the transitional and rarefied flow regimes. Several structural analysis codes were investigated, with LS-DYNA [25] producing the most consistent solutions for thin-filmed materials. A static aeroelastic analysis was performed on two geometries for the continuum and transitional flow regimes using the low and high fidelity approach. An example of these results is shown in Fig. 3, where the structural response is shown to the left and the flowfield is shown to the right. A low fidelity flutter analysis was performed, requiring a time coupling approach. The suggestions for future

work included the addition of high temperature aerothermal effects, and the coupling of a thermal response analysis. In addition, a high-fidelity, dynamic, aeroelastic analysis was suggested.




(a) (b)

Figure 3: IAD configuration for FEA and CFD for a contiuum simulation [93]. The structural dynamics solution computed using LS-DYNA is shown in (a), while the flowfield solution computed using NASCART-GT is shown in (b).

Lee [68] expanded the inviscid capability of NASCART-GT to include thermochemical nonequilibrium. With this capability, the IAD configuration (referred to as a ballute in [93]) explored by Rohrschneider was re-analyzed. An example flow solution can be seen in Fig. 4. A static aeroelastic analysis was performed using this aerodynamic solution, coupled with LS-DYNA.

In addition to a series of experimental work, Clark [23] used CFD to compute steady-state flows over a supersonic tension cone decelerator. The work presented inviscid results using NASCART-GT, and laminar viscous results using the overset capabilities of the structured solver, OVERFLOW [18]. The inviscid calculations yielded accurate bow shock shape and location, as well as fore-body pressure, but demonstrated larger inaccuracies in the aft-body region. The wake of the tension cone is heavily governed by viscous flow phenomena that cannot be captured by an inviscid solver. The viscous analysis showed that the wake region could be captured with much higher accuracy, including near perfect agreement in

the aft-body pressure distributions for Mach numbers up to 3 and angles of attack up to $20^o$. However, it was suggested that higher Mach numbers and angles of attack likely led to transition to a turbulent boundary layer on the leeward portion of the tension cone. The computational work only studied the flow about the final deformed shape in order to characterize the level of fidelity required. Suggested future work consisted of a comprehensive validation of structural dynamics codes, with the goal of coupling the fluid dynamics and structural dynamics for static validation of supersonic IAD configurations.

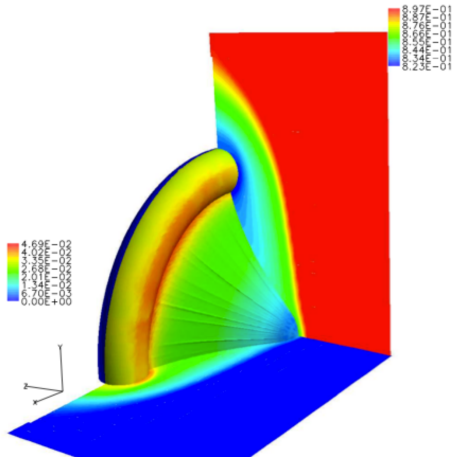


Figure 4: Example flow solution presented by Lee [68], in which chemical nonequilibrium has been accounted for. $N_2$ mole fractions are shown on x-plane and N mole fractions on y-plane.

The work presented by Tanner [107] aimed to address these issues. The structural dynamics was solved using LS-DYNA, based on the previous studies by Rohrschneider [93], demonstrating the applicability of LS-DYNA for thin-filmed membranes. An extensive analysis was performed using LS-DYNA with the purpose of understanding the most optimal input options for the current application. Analyses were compared with theoretical results in order to provide a set of best practices. The aerodynamic analysis used FUN3D, an unstructured Navier-Stokes solver, developed at NASA Langley Research Center. The motivation behind choosing this code was its ability to accurately compute viscous flow features in the wake, adaptively align the mesh to the flow gradients, and incorporate various turbulence models. Additionally, there exists the capability of mesh deformation

based on solid body movement. The two analysis codes were loosely coupled to produce steady-state FSI simulations, and the process consisted of running each solver to convergence before iterating on the coupling process.

Gidzak [41] has demonstrated a partitioned, tightly coupled method between the unstructured, finite volume solver, US3D [87, 31], and an explicit finite difference membrane solver. The structural solver was developed specifically for his thesis work, since any of the tested finite element solvers presented difficulty in computing the parachute dynamics. This approach required a method for performing grid deformation, based on structural displacements. A DGB parachute configuration was analyzed for MSL entry using a static parachute that was allowed to pitch and yaw in a two-dimensional plane. The pressure and viscous forces acting on the parachute surface drove this two degree of freedom motion. The grid resolution in the turbulent wake of the MSL vehicle played a key role in capturing the unsteady flowfield that drives the parachute motion. Additionally, small time steps were required in order to capture the pressurization cycle, leading to long computational times to resolve the parachute dynamics. A tension cone IAD was also investigated for a notional Mars entry. The simulation accounted for the interaction between the fluid flow, structural dynamics, and used a 1-D heat transfer model at the surface. It was found, for the analyzed conditions, that there would be no material failure due to stress or thermal loading. It was noted that the grid deformation approach only works for deformations that are not too large, as well as for cases in which there are no joining or splitting of surfaces. In that case, re-meshing would be required, which can often be computationally expensive, and introduces concerns with respect to interpolating onto the new grid. Additional areas of concern that were raised include the effects of varying conditions along the entry trajectory. A comparison between the start and end times of an FSI simulation may correspond to largely varying conditions along the trajectory. Additionally, the deceleration of the vehicle introduces a changing reference frame that theoretically should be accounted for.

The analysis of a supersonic disk-gap-band parachute was presented by Karagiozis, et al. [56] as a demonstration of coupling between a Large-Eddy Simulation (LES) and a structural membrane solver. Simulations were computed for conditions corresponding to

wind tunnel experiments performed by NASA [100]. The configuration of interest was a disk-gap-band parachute attached to an aeroshell capsule as seen in Fig. 5. The flow was modeled using the compressible LES equations on a Cartesian mesh with adaptive mesh refinement (AMR). The structural dynamics was modeled using Kirchohff-Love thin-shell theory, treating the structure as a membrane with non-zero bending stiffness. The fluid dynamic boundary conditions were enforced using the ghost-fluid-method (GFM) [36] and the level set method [21, 28] to represent the solid surface. Slip wall boundary conditions were applied, with boundary layer approximations to the Reynolds Averaged Navier-Stokes (RANS) equations. It was re-iterated that many challenges still exist in predicting the location of transition from laminar to turbulent flow. The results for each case presented in reference [56] required approximately four months of run time on 100 CPUs, highlighting the expensive computational cost associated with accurately resolving the turbulent wake structure in an LES analysis of a trailing parachute configuration.

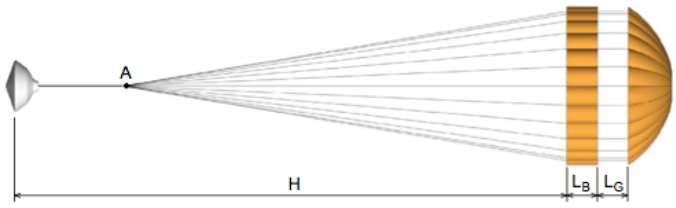



Figure 5: Disk-Gap-Band parachute configuration used in the LES FSI analysis [56].

The inflation/deflation cycle of the tension cone configuration was examined by the same research group in Ref. [55]. The membrane structure was treated as a thin shell, loosely-coupled to the fluid solver. The results that were presented were only preliminary, but demonstrated non-linear interactions through the pressurization cycle. The simulations contained complex physics that is difficult to validate, but the work demonstrated progress towards that end.

Another set of LES simulations of a tension cone pressurization cycle was studied by Kramer et al. [62]. A staggered approach was taken to couple the LES solver with a thin shell solver. Simulations included the analysis of the start-up transients, as well as the

fully inflated configuration. The pressurization cycle was compared to experimental data using force and deflection measurements. A simulation was also run, which investigated the possibility of the system being driven into resonance. The resonance comes from a matching between the fluid and structure fundamental frequencies in the post shock region.

The work by Gilmanov et al. [43] presented an FSI framework, which coupled a Cartesian solver with a particle-based meshless structural method. The Cartesian solver utilized a hybrid immersed boundary method capable of resolving all geometry features. Simulations were performed on a sphere and a tension cone configuration.

With the goal of developing a fast framework for FSI simulations, Refs. [59, 60] show the development of a loose-coupling between two simplified physics models. The aerodynamics was computed using the Configuration Based Aerodynamics (CBAERO) software package, which uses surface inclination methods to approximate supersonic pressure distributions. A simplified, flexible membrane solver was implemented into the software, now called CBAERO-FLEX. Results were presented for a tension cone IAD. Despite the fact that complex physics cannot be accurately predicted, the tool is very fast and useful for preliminary design analysis.

Additional simulations of planetary entry configurations can be considered with a specialized case of FSI. That is, one in which the geometry configuration is not elastic, but is allowed to undergo rigid body motions. The study of atmospheric entry probes by Murman et al. [82] examined free and forced oscillations, with the goal of characterizing the aerodynamic coefficients. Several entry configurations were considered, including the Viking, Genesis, and Mars Exploration Rover capsules. The work by Stern et al. [105] studied the free flight dynamics of the MSL capsule. Simulations were run where the capsule was provided an initial release angle, and the geometry was allowed to respond to the aerodynamic loads. Several release angles were examined, and aerodynamic coefficients were computed based on an aerodynamics model. The computational data was compared against ballistic range data from Ref. [98].

The coupling between separate fluid dynamics and structural dynamics solvers requires a coupling algorithm. In a series of work primarily including Charbel Farhat, staggered

coupling schemes were developed for the use in partitioned FSI problems. In Refs. [33, 70], the Conventional Serial Staggered (CSS) algorithm is discussed, and presents further development for a parallel version. The CSS algorithm is a simple algorithm that sequentially passes data back and forth between solvers. The CSS method is a straight forward approach for coupling two disciplines, but is only $1^{st}$ order accurate at best, regardless of the order of accuracy of the fluid and structural solvers. Accuracy can be improved by performing sub-iterations with the algorithm, with a large computational cost associated with it. The Improved Serial Staggered (ISS) algorithm was developed as a means to mitigate this issue. The algorithm has a similar computational complexity as the CSS, is sub-iteration free, and demonstrates improved numerical properties. Their results included a flutter analysis of the AGARD Wing 445.6, demonstrating the superiority of the improved method.

Further developments on the serial staggered algorithms appear in Refs. [40] and [35], where two schemes were developed, with provably $2^{nd}$ order accuracy. The schemes were referred to as the Generalized Serial Staggered (GSS) method. The applications of the method presented in the paper include the 3-D analysis of a complete F-16 fighter jet. Farhat et al. [34] continued to make the case for partitioned coupling methods, by demonstrating that carefully constructed, staggered coupling schemes can still achieve numerical stability and provable accuracy, without the need for sub-iterations. They demonstrated this fact by studying compressible flow interactions with a submerged cylindrical shell in water.

A high order coupling scheme applied to moving meshes was discussed in Ref. [112]. The algorithm was used for partitioned schemes, and used implicit and explicit (IMEX) Runge-Kutta steps. In their work, the time integration of each individual solver was performed using high order accurate implicit schemes, while the coupling at the interface was done using an explicit approach. They studied a 1-D piston problem, and demonstrated that their approach was able to maintain the high order of accuracy of the implicit and explicit schemes. In addition, they showed that the same accuracy could be achieved as a $2^{nd}$ order accurate staggered scheme for less computational work.

The work by Froehle and Persson [38] examined a high-order accurate coupling scheme

applied to a flapping flight simulation. The scheme demonstrated up to $5^{th}$ order accuracy, coupling the Navier-Stokes equations using a Galerkin arbitrary Lagrangian-Eulerian approach with a structural solver using a continuous Galerkin approach. The partitioned coupling scheme was based off of the IMEX scheme. The paper demonstrated that it is possible to achieve higher order accuracy without the use of sub-iterations.

An important consideration regarding partitioned schemes is the role that added mass effects can have. The concept of added mass, including its physical and mathematical formulation is discussed in Refs. [20, 111]. The effects of this term become relevant when the ratio of fluid mass to structural mass becomes large (i.e. parachutes in air). The term "added mass" refers to the artificial change in the inertia term describing the flexible structure. For partitioned coupling schemes, the fluid pressure is applied to the structure at a time that is offset from the actual physical time of application. Since the fluid mass is much greater than the structural mass, this fluid mass can dominate in the inertia term of the structural equations of motion. For incompressible flows, the added mass is a function of fluid density and domain/interface length scales. For compressible flows, the added mass is a function of fluid density, speed of sound, and time. Thus, for compressible flows, the added mass is only felt within a small region of the interface, with the distance depending on the speed of sound and the time step. This implies that the magnitude of the added mass reduces as the time step reduces. Indeed, for compressible flows, sub-iteration methods can be stable for any fluid density and structural mass, provided that the time step is small enough. However, for incompressible flows, the sub-iteration method can be unstable regardless of how small the time step is.

The work in Ref. [9] studied a 1-D elastic piston by investigating staggered algorithms and monolithic approaches. The staggered algorithm used separate numerical solvers, augmented by a predictor-corrector approach. This introduced a time lag between the fluid and structural systems, which led to deviations in the mean piston position, as well as non-physical energy production. The results were compared to a monolithic coupling approach that eliminated both of these inaccuracies. Their work concluded with a discussion of the Interaction Consistency Law. This law requires that "the time dependence of the boundary

conditions for the fluid and structure solvers has to be consistent with the discrete time integration of the structure and fluid solvers, respectively" [9]. Essentially, this implies that for staggered coupling schemes, where the application of boundary conditions imposes a time lag between systems, there will be an inconsistency between solvers. This inconsistency manifests itself as a non-physical energy production, which can lead to instability and inaccurate mean positions of the piston.

This problem was also studied by Jaiman et al. in Ref. [53], where a new loosely coupled staggered algorithm was presented. The method was called the combined interface boundary condition (CIBC), and was demonstrated by considering the open and closed domain elastic piston problems. The closed domain problem was used to study numerical stability, while the open domain was used to characterize damping rates of the numerical scheme. Further comparisons between staggered and monolithic approaches have been made by Ref. [78], using the 1-D elastic piston as a model problem. The work demonstrated the increased error in partitioned schemes due to artificial energy production compared to the more stable monolithic approach.

Reference [92] investigated fluid-structure interactions for the application of aeroacoustic instabilities in solid rocket motors. The baseline test problem was that of an elastic piston. However, the problem was simplified further by considering the fluid domain to be of zero dimensions. The flow variables in the adiabatic fluid chamber were assumed to be uniform, and that the piston compression was done so reversibly. This simplification provides an analytic solution that takes on a relatively simple form. Artificial energy production was characterized, and multiple predictor schemes were presented in order to demonstrate the stability of the simulation.

In the work by Piperno [89], two coupling schemes were discussed. The schemes were shown to be $2^{nd}$ order accurate by using a structural predictor. This allows momentum to be conserved at the interface, despite the fact that there is still an inconsistency in continuity. The scheme is studied by examining the supersonic flutter of a flat plate.

Up to this point, the literature review has been focused on both the general EDL problem, and FSI coupling strategies. However, the focus of the current research utilizes several

numerical techniques that require additional overview. First, the use of Cartesian grid techniques is discussed, followed by a review of surface boundary condition methodologies relevant to numerical simulations where geometries are immersed within a Cartesian mesh. Additional research is discussed as it is relevant to the Cartesian grid CFD approach.

CFD solvers require a discretized spatial domain, in the form of a grid. Grid topologies can vastly differ, depending on the primary objective of the CFD solver, and a cost benefit analysis of the strengths and weakness of each type is necessary. Two important aspects to consider are the data connectivity of the mesh and the way in which solid surfaces are represented.

The data connectivity refers to the way in which individual nodes and computational elements are spatially arranged. Grids can be classified as either structured or unstructured. Structured meshes have an ordered connectivity, and are designed such that the mesh in physical space can be mapped to a uniform mesh in computational space. Every node can be identified by an ordered pair of indices corresponding to each spatial direction. The advantages of using such a mesh is that connectivity is implicit in the way the data is stored, and no additional memory is required to store this information. This has the additional effect of significantly reducing computational time. However, the primary downside is that it is difficult to create structured meshes for complex geometries, or configurations containing multiple geometries in close proximity. Examples of structured grid solvers that are used for EDL applications include the two NASA codes, DPLR [116] and LAURA [45]. The first steps towards extending DPLR to handle FSI problems have been made with regards to dynamic mesh motion, along with the addition of the appropriate flux components, and enforcement of the Geometric Conservation Law [6].

Unstructured meshes, on the other hand, are very flexible when utilized with complex geometries. They are unstructured in that there is no ordered connectivity to the data. This means that nodes and computational elements can be added wherever necessary for geometry or flow resolution. This also allows arbitrarily shaped elements to be used, which are only limited by the robustness of numerical algorithms. The downside to such approaches is that additional memory is required to store element connectivity. The expense associated

with determining the connectivity may only be a one time initialization cost, unless the mesh connectivity is modified during a simulation. Examples of unstructured grid solvers used for EDL applications include FUN3D [3] and US3D [88].

The second mesh characteristic of importance is the way in which the surface boundary is represented, depending on whether it is body-fitted or not. Body fitted meshes are designed such that any physical geometry represented in the mesh is done so by requiring grid points to lie on the surface of the geometry. These nodes, in essence, define the physical geometry as represented by the computational mesh. This approach is often the most accurate because the geometry can be discretized to very fine detail. However, the difficulty comes in mesh generation, when enforcing the requirement that the mesh nodes align with the geometry. The four codes previously mentioned (DPLR, LAURA, FUN3D, and US3D) all use body-fitted meshes.

A non body-fitted mesh is one in which the physical geometry is immersed within either a structured or unstructured mesh. The geometry intersects the mesh at arbitrary points, allowing for the automatic mesh generation of complex geometries, including multi-body configurations. The local information near the intersection of the physical geometry and mesh must be used in order to enforce boundary conditions. The enforcement of boundary conditions is the most challenging aspect of using this approach.

Since the present work utilizes an unstructured, Cartesian, non-body fitted approach, further discussion on the boundary treatment approach follows. Mittal and Iaccarino [80] and Dadone [27] provide an extensive review of immersed boundary methods. The boundary is represented by supplying the fluid state for cells near the surface, such that the boundary condition is satisfied. Two approaches can be broken down into continuous forcing methods and discrete forcing methods. The continuous forcing approach modifies the governing equations to take into account the effect of the solid surface, while discrete forcing leaves the governing equations unaffected, but the discretization is adjusted near the surface to take the boundary into account.

Continuous forcing is further broken down into elastic and rigid boundaries. Elastic boundaries, such as the muscle fibers of the heart (Peskin 1972, 1981) use a forcing function

with a relation such as Hooke's Law. The forcing function is represented by a Dirac delta function. However, it is numerically described by smoother functions, primarily because the grid does not necessary line up with the surface. The choice of the distribution function is a key component of such methods. Rigid bodies must be treated differently by using a very large artificial stiffness, and using a constitutive law relating the force to a displacement from an equilibrium position.

The discrete approach can be broken down into indirect and direct methods. The indirect method uses the numerical solution to obtain a forcing term. The approach removes user specified parameters, and provides favorable stability characteristics. The direct approach modifies the computational stencil near the surface in order to specify the boundary condition. The finite difference ghost cell approach uses ghost cells in the numerical stencil to specify state in order to enforce the boundary condition. The state of the ghost cell is set by a variety of interpolation methods using neighbor cell data, as well as local surface properties. Discrete forcing allows for sharp interface representation, as opposed to the continuous forcing that introduces smoothing.

When applied to moving surfaces, the discrete approach provides a sharp geometry representation and a sharp boundary condition representation. The method naturally extends to moving surfaces, with the added complexity of dealing with "freshly-cleared" cells. Cells that emerge from inside the geometry over the course of a time step result in a discontinuity in time, since there was no valid fluid state initially. Emerging cells can either be temporarily merged with nearby cells, or interpolated from neighboring data.

A novel immersed interface method for incompressible Navier-Stokes has been developed by Brehm [17, 13]. The method was extended to moving boundaries [14] and compressible flows [15, 16]. The boundary conditions form a sharp interface method that is able to represent high order accuracy at the boundaries. These methods have also been extended and implemented within the CFD code, LAVA (NASA Ames) [61].

The cut-cell finite volume approach is able to provide conservation at the immersed surface by cutting cells such that the boundary is resolved in a similar topology to body-fitted meshes. Cells are re-shaped, such that the final outcome results in non-Cartesian cells

at the surface that become body-fitted. A cut-cell method has been demonstrated for FSI problems in Refs. [48, 97], where time dependent force oscillations have been addressed. Forrer and Berger [37] discuss a compressible Euler, Cartesian cut-cell approach, which addresses the small cell problem that often leads to numerical instabilities. Moving geometry simulations are shown with discussions on numerical accuracy. Simulations include a moving wall with constant and varying speeds.

Some Cartesian methods have taken an approach similar to overset methods, where there may be two overlapping Cartesian meshes. The mesh near the surface remains unchanged throughout the simulation. Instead, it travels through a second, stationary mesh. The work of Karimian and Ardakani [57] use the multi-zone approach of Mirsajedi et al. [79], and compare stationary to relative motion simulations.

An immersed boundary method for simulating FSI on Cartesian meshes is presented by Harris [49] and Harris and Kolobov [50]. The immersed boundary method is used, along with Lagrangian marker points to track the surface. The solver is best suited for incompressible, low Reynolds number flows. Results are presented for a plunging airfoil and a flapping flexible wing.

A large body of work related to FSI problems and immersed boundary methods has come out of the Farhat Research Group at Stanford University. An embedded boundary method is used in the work presented by Wang, et al. [115], where the velocity and pressure boundary conditions are treated by an exact Riemann solver at the interface. The method is extended to viscous problems in [32]. The method is further extended to compressible, turbulent flows in [64]. The development of surface boundary conditions that are solved by a Riemann solver have been shown to possess high order accuracy. Additionally, there is a discussion on a fast algorithm for determining the nearest wall point for any computational cell, as well as loads transfer calculations for FSI problems.

An interface tracking method for FSI problems is presented by Wang et al. [113, 114], which is able to accurately track geometry movements. Two methods are presented, with one based on a projection approach and the other on a collision approach. The former is significantly faster, but requires the geometry to form a closed surface. The second method,

however is able to track an interface for non-closed surfaces. This is highly beneficial for many parachute and decelerator configurations.

Lakshminarayan and Farhat [63] describes an approach referred to as an ALE-Eulerian method, with the goal of allowing Eulerian grid features to maintain approximate locality to the geometry as it moves (without performing grid adaption). Strictly Arbitrary Lagrangian Eulerian (ALE) approaches for FSI problem rely solely on deforming the mesh either rigidly or elastically. Since a strictly ALE approach is typically used for body-fitted meshes, the approach only works well for small deformations. Large deformations lead to grid lines crossing. On the other hand, a purely Eulerian approach using the embedded boundary method allows the geometry to move through the stationary mesh. However, regions of high grid resolution do not automatically move with respect to the body. A specific example cited is the high mesh resolution required to capture the viscous boundary layer around a geometry. As the geometry moves and deforms, the boundary layer region of the grid does not follow suit. Thus, the idea of the approach is to decompose structural motion into a rigid motion component and a relative motion component, such that the grid moves to maintain high resolution in the boundary layer. The rigid motions are captured using an ALE formulation, while the local, small deformations are captured using the embedded boundary methods. This method is not limited specifically to the embedded boundary method, but can be used for immersed boundary methods, immersed interface methods, or other similar approaches.

The work by Muralidaran and Menon [81] presents a cut-cell methodology for simulation of flow interactions with structures. A simulation is presented for 1-D isentropic flow, which allows for the measurement of global accuracy of a smooth solution. Additionally the conservative nature of the scheme is verified by demonstrating a zero loss in mass. A block structured grid approach is used, along with adaptive mesh refinement (AMR). Moving body simulations include the flow between moving walls, as well as the reacting flow generated by the detonation of a cylinder due to impulsive motion.

Solution-based mesh adaption applied to moving body problems is shown in Ref. [104]. The results first show unsteady adaption on a stationary case. Next, a NACA 0012 airfoil

undergoing rigid body motion is studied. For the implementation, only refinement was included (no coarsening). The results demonstrate improved accuracy when solution-based mesh adaption is applied to moving body problems.

## *1.3 Goal & Objectives*

The goal of this work is to develop and validate a time-accurate aeroelastic methodology, utilizing an unstructured Cartesian immersed boundary method, that can be used for the analysis of aerodynamic decelerators. This goal is achieved by addressing several objectives that lay the foundational groundwork.

The first objective is to perform verification and validation of the CFD solver, utilizing an unstructured Cartesian immersed boundary method, for stationary and non-stationary simulations. The objective is to first ensure that the CFD solver sufficiently captures the necessary physics for stationary atmospheric entry configurations. In addition, the accuracy of the numerical schemes and boundary conditions must be examined for non-stationary configurations. This is done by studying 1-D and 2-D prescribed motion simulations. The effects of surface velocity and mesh resolution are evaluated, and conclusions are drawn, which allows extension to FSI problems.

The second objective is to develop an FSI methodology that fills a mid-fidelity gap in the computational analysis of atmospheric entry configurations. The use of an unstructured, Cartesian CFD solver provides several benefits towards this end. Grid generation is performed without the need for manual user intervention. This also extends to the use of solution-based mesh adaption, and grid modifications that occur as the structure undergoes deformations. These are demonstrated through the simulation of prescribed motion, six degree-of-freedom motion, and deformable FSI problems.

The third objective is to demonstrate time accurate fluid-structure interaction simulations of the dynamic motion of a tension cone. Several validations are first considered prior to studying the tension cone. The first problem that is studied is a 1-D elastic piston interacting with fluid in a chamber. This provides a relatively simple physical problem to study the numerical characteristics of the time accurate coupling. Supersonic flow past a

deformable panel is also studied to validate the methodology for relevant flight conditions and dynamic response. Finally, the time accurate analysis of a tension cone IAD is performed, which demonstrates the capability of the developed work to perform dynamic FSI simulations.

# CHAPTER II

# GOVERNING EQUATIONS AND NUMERICAL IMPLEMENTATION

The mathematics describing fluid-structure interactions consists of a combination of two distinct sets of governing equations, which describe the fluid dynamics and structural dynamics. The fluid dynamics is constrained to the domain, $\Omega_f$, which is typically made up of the relevant flow regime surrounding a flight vehicle, or other body of interest, while the structural dynamics takes place on the domain, $\Omega_s$, and only consists of the geometry of interest that is subject to structural deformation. The physical problems are joined at the interface between both domains, $\Gamma_{fs}$. The governing equations of such physics are discussed in this chapter, followed by a detailed presentation of the numerical implementation of the FSI framework.

## *2.1 Governing Equations for Fluid-Structure Interactions*

The governing equations for fluid-structure interactions consist of the Navier-Stokes equations describing the fluid dynamics, and the equations of motion for structural dynamics. The coupling between the two takes place at the domain interface boundary conditions.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho u_i\right) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}\left(\rho u_j\right) + \frac{\partial}{\partial x_i}\left(\rho u_j u_i\right) + \frac{\partial p}{\partial x_j} = \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) + \lambda \frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \tag{2}$$

$$\frac{\partial}{\partial t}\left(\rho E\right) + \frac{\partial}{\partial x_i}\left[\rho H u_i\right] = \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) + \frac{\partial}{\partial x_i}\left[u_j \mu\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) + u_j \lambda \frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \tag{3}$$

$$\rho \frac{\partial^2 u_j}{\partial t^2} = \frac{\partial}{\partial x_i}\left(\sigma_{ij}\right) + \rho b_j \tag{4}$$

Equations 1 - 3 make up the Navier-Stokes equations, more specifically the continuity, momentum, and energy transport equations. The equations are solved in the fluid domain where $\rho$ is the fluid density, $u$ is the velocity, $\mu$ is the fluid viscosity, $\lambda$ is the bulk viscosity, $\delta_{ij}$ is the Kronecker delta, $E$ is the total specific energy, $H$ is the total specific enthalpy $\left(H = E + \frac{p}{\rho}\right)$, $p$ is the pressure, $T$ is the temperature, and $k$ is the thermal conductivity. In addition to the three transport equations, a thermodynamic relation is needed such as the perfect gas equation to close the system. In all work herein, the gas is assumed to be calorically perfect.

Equation 4 is the governing equation for structural dynamics in the domain, $\Omega_s$, where $\rho$ is the material density, $u$ is the displacement, $\sigma_{ij}$ is the stress tensor, and $b$ is the vector of body forces. In addition to the equation of structural dynamics, a material constitutive relation is also necessary to relate stress and strain.

The boundary conditions are specified at the interface, and must satisfy three conditions. The fluid stress must match the structural material stress, the position of the interface must match between the two sets of equations, and the interface velocities must match.

## *2.2 Geometric Conservation Law*

The vast majority of partitioned, FSI solvers described in the literature use a three-field approach: 1) fluid equations, 2) structural equations, and 3) deforming mesh motion equations. The reason for this third equation is that for body-fitted mesh simulations, the moving interface requires the mesh to deform. Locally near the interface, the mesh must deform in accordance with the structural motion, while adjusting at a distance to maintain sufficient grid quality. In the case of non-body-fitted, Cartesian meshes, there is no need to solve the deforming mesh equations since the structure moves through a stationary mesh. However, in the case of the ALE formulation (where the mesh undergoes rigid body motion), the mesh nodes are moving, while maintaining the same shape for all computational cells.

As the mesh deforms, enforcement of the Geometric Conservation Law (GCL) helps improve numerical stability. The GCL states that the flow solver must be able to predict

uniform flow conditions, even when the mesh is undergoing deformation. The mathematical description of this statement yields

$$\frac{d}{dt}\int_V dV = \int_S \vec{W}_s \cdot d\vec{S} \tag{5}$$

where $V$ is the computational volume, $S$ is the computational surface area with the direction based on the outward unit normal, and $\vec{W}_s$ is the velocity of the computational surface. This equation states that the change in computational volume over a given time interval must be equal to the total volume swept by the computational boundary.

The volume of any given computational cell is easily known based on the geometry of the cell. Additionally, the cell nodal velocities are known based on the movement of the mesh over a time step. However, the numerical integration requires knowledge of the face velocity as a whole, and not in terms of the nodes. Thus, the face velocity can be approximated by averaging the known nodal velocities. In the case of generalized mesh deformation, the volume swept by the cell faces using approximate velocities is not necessarily equal to the change in geometric volume of the cell. The GCL can be satisfied by either modifying the cell volume, or by considering a modification to the face velocities. However, as previously noted, no special considerations are required for the present numerical formulation.

### *2.3 Fluid Dynamics*

The Navier-Stokes equations (Eqns. 1-3) are often written in vector form, as this is more convenient for numerical discretization. The equations are written in conservation form, where the fluxes are split into convective (inviscid) and dissipative (viscous) components. The governing equations are written in a similar form as found in Ref. [106], with the extension to an ALE formulation.

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial}{\partial x}\left(\vec{E} - \vec{E_v}\right) + \frac{\partial}{\partial y}\left(\vec{F} - \vec{F_v}\right) + \frac{\partial}{\partial z}\left(\vec{G} - \vec{G_v}\right) = \vec{S} \tag{6}$$

The conservative state vector ($\vec{U}$) defines the fluid state, and the source term ($\vec{S}$) can represent any number of source terms added to the governing equations. Examples of source

terms include body forces, chemical rate production terms, turbulence production terms, or the axisymmetric production term (as shown below).

$$\vec{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{bmatrix} \qquad \vec{S}_{axi} = \frac{1}{r} \begin{bmatrix} 0 \\ 0 \\ -p \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

The flux vectors are split into inviscid fluxes, $\vec{E}$, $\vec{F}$, $\vec{G}$, and viscous fluxes $\vec{E_v}$, $\vec{F_v}$, $\vec{G_v}$.

$$\vec{E} = \begin{bmatrix} \rho\left(u - \dot{x}\right) \\ \rho u\left(u - \dot{x}\right) + p \\ \rho v\left(u - \dot{x}\right) \\ \rho w\left(u - \dot{x}\right) \\ \rho e\left(u - \dot{x}\right) + pu \end{bmatrix} \quad \vec{F} = \begin{bmatrix} \rho\left(v - \dot{y}\right) \\ \rho u\left(v - \dot{y}\right) \\ \rho v\left(v - \dot{y}\right) + p \\ \rho w\left(v - \dot{y}\right) \\ \rho e\left(v - \dot{y}\right) + pv \end{bmatrix} \quad \vec{G} = \begin{bmatrix} \rho\left(w - \dot{z}\right) \\ \rho u\left(w - \dot{z}\right) \\ \rho v\left(w - \dot{z}\right) \\ \rho w\left(w - \dot{z}\right) + p \\ \rho e\left(w - \dot{z}\right) + pw \end{bmatrix} \tag{8}$$

$$\vec{E_v} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xx}u + \tau_{xy}v + \tau_{xz}w - q_x \end{bmatrix}$$

$$\vec{F_v} = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yx}u + \tau_{yy}v + \tau_{yz}w - q_y \end{bmatrix} \tag{9}$$

$$\vec{G_v} = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zx}u + \tau_{zy}v + \tau_{zz}w - q_z \end{bmatrix}$$

The governing equations must be closed using a thermodynamic state relation. In the case of a perfect gas,

$$p = \rho RT \tag{10}$$

A viscosity model is also required, with Sutherland's Law being a common choice for calorically perfect gases within a valid temperature and pressure range. Additionally, the thermal conductivity can be related to the viscosity and specific heat capacity through the Pandtl number, which is approximately constant for the previously stated assumptions.

Mesh motion is taken into account by considering the Arbitrary Lagrangian Eulerian (ALE) formulation, where $\vec{x}$ is the velocity of the reference domain with respect to the global frame. Reference [30] provides a comprehensive review of ALE methods, and how they are formulated. Typically, the Navier-Stokes equations are presented in an Eulerian formulation as this provides the most straight forward approach for numerical solutions. In order to account for mesh motion, a Lagrangian perspective is taken with respect to the mesh coordinates. The relative motion of the mesh leaves the governing equations largely unchanged, with the effects of relative motion absorbed into the convective terms and boundary conditions.

In finite volume methods, a correction term to the cell volume is required in order to take into account volume changes caused by mesh motion. This correction is generally written as

$$\vec{U_c} = \vec{U_o}\left(1 - \frac{\Delta Vol}{Vol_f}\right) \tag{11}$$

where $\vec{U}_c$ is the required correction in the conserved state vector, $\vec{U}_o$ is the conserved state vector prior to the correction, $\Delta Vol$ is the change is cell volume, and $Vol_f$ is the final cell volume. This correction, however, is not necessary in the present work because of the fact that the computational cells do not undergo volume changes, while the mesh is subject to rigid body motions.

## *2.4 Structural Dynamics*

The governing structural dynamics equations [4] can be written in matrix form, which is typically used when applying a numerical discretization.

$$[M]\,\ddot{\vec{u}} + [C]\,\dot{\vec{u}} + [K\,(\vec{u})]\,\vec{u} = \vec{F} \tag{12}$$

where $[M]$ is the mass matrix, $[C]$ is the damping coefficient matrix, $[K\,(u)]$ is the stiffness matrix, $\vec{F}$ is the load vector, and $\vec{u}$ is the displacement vector.

Non-linearities enter this system of equations either through large displacements, or through material non-linearities in the stress-strain relation. The matrices must be constructed by using an element formulation that can take various numerical forms. The solution to the system of equations typically requires inversions of large matrices, and a time marching technique to integrate in time.

## *2.5 Fluid-Structure Coupling Methods*

The governing equations for fluid-structure interactions can be numerically coupled in various ways, providing a range of analysis fidelity and computational cost. The two primary approaches can be broken into "direct" and "partitioned" methods. Direct methods refer to approaches where the two sets of governing equations are formulated in a way such that they can be solved simultaneously, whereas partitioned methods treat the governing equations separately so they can be individually solved by different numerical approaches.

The direct method is the most physically accurate approach since both physical fields are solved simultaneously. Direct methods can also be referred to as "fully coupled", "monolithic", or "fully implicit". The full coupling of this method is very useful for the accurate

simulation of time dependent problems. In certain situations the physical time scales may be so tightly coupled that direct methods are the only viable options. An advantage to this method is that the interface boundary conditions are maintained in a more natural sense, due to the inherent dependence built into the complete set of equations. Despite this attractiveness, the challenges associated with such an approach often outweigh the benefits. The required discretizations are often quite different between the fluid dynamics and structural dynamics. This includes both the physical meshing, as well as the appropriate time scales. In addition, the numerical procedures for solving the two physical problems are so different, that it is challenging to develop numerical methods to solve the entire system simultaneously. Finally, in a practical sense, the development of such software becomes extremely difficult due to the size of the code base.

The alternate approach to direct methods is to use partitioned methods. They are partitioned in the sense that each physical discipline being solved is done so separately, as depicted in Fig. 6. In regions where the physical problems interact, boundary conditions must be imposed that allow for physically accurate coupling. Unlike the direct method, there is no immediate response between the fluid and structure due to the inherent time lag caused by partitioning. However, the fact that different solvers can be used for each separate physical model implies that there are more options in optimizing each. As the interface boundary conditions must be periodically exchanged between solvers, the rate at which this is done plays an important role in the overall time accuracy of the coupling scheme. At a minimum, data must be transferred at a rate faster than the smallest time scales, if a time accurate scheme is desired. In the situation where the time scales between the fluid dynamics and structural dynamics differ significantly, sub-cycling can be used to increase efficiency. Sub-cycling is a methodology that allows different time steps to be taken within each solver (at its most efficient value), which also implies that each solver will proceed for a different number of numerical iterations. For example, if the structural time scale is an order of magnitude larger than the fluid dynamics time scale, this can reduce the computational cost of the structural dynamics calculations by an order of magnitude.

Partitioned schemes, though not fully coupled, can still provide a range of coupling from

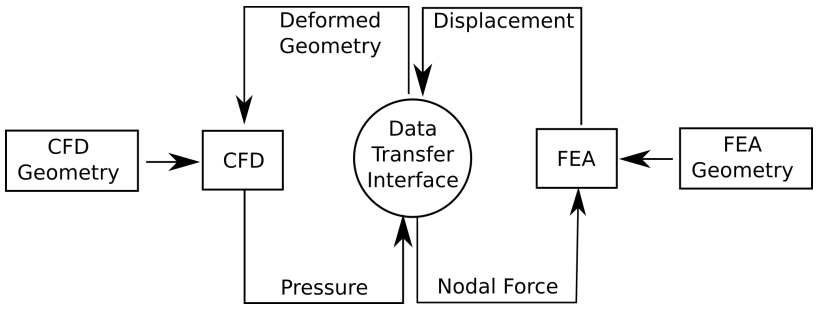



Figure 6: Partitioned coupling strategy, where the fluid dynamics (CFD) and structural dynamics (FEA) are computed using separate solvers.

weak to strong. The weakest form of coupling can be applied to steady-state analyses for which the final solution is independent of the time history. There is no requirement on time accuracy within each solver, nor ensuring that each solution has advanced the same physical time. The most common approach is to run each solver to convergence before transferring data. A stronger form of partitioned coupling maintains time accuracy by advancing and transferring data at a rate sufficient to satisfy the physical time scales. This can be accomplished by transferring boundary condition data at every time step, or using the concept of sub-cycling described above. The time accuracy of this approach is highly dependent on the coupling scheme that is used.

The coupling schemes that are used in the current methodology fall under a class of schemes described by Farhat et al. [35] as the generalized serial staggered method (GSS). The GSS method provides several formulations, depending on chosen control parameters, as well as how the time steps are synchronized between the fluid dynamics and structural dynamics.

The first form of the scheme simplifies to the common approach often referred to as the Conventional Serial Staggered (CSS) method. This method is at best $1^{st}$ order accurate in time, but is widely used due to its ease of implementation. It is, however, susceptible to the negative effects of artificial viscosity, and to undesirable stability characteristics. The scheme can be broken down into a set of serial tasks:

1. update the fluid dynamic mesh using $x^{n+1} = x^n + M^n \Delta u_\Gamma^n$ or initialize the mesh at time $t^n$ if first time step

2. use this geometry configuration to update the fluid state from time $t^n$ to time $t^{n+1}$

3. apply the fluid dynamic pressure loading ($p^{n+1}$) as nodal forces ($f^{n+1}$) in the structural solver

4. solve the structures system from time $t^n$ to time $t^{n+1}$

where $M^n$ accounts for the mapping between meshes, is taken at time $t^n$, and is assumed to be of value $M^n = 1$ when applied to simulations where the CFD mesh is identical to the FEA mesh. $x$ refers to the generalized coordinates of the structure, and $\Delta u_\Gamma^n$ is the change in structural displacement from time $t^{n-1}$ to time $t^n$, $\Delta u_\Gamma^n = u_\Gamma^n - u_\Gamma^{n-1}$.

The CSS scheme is depicted in Fig. 7a. One advantage of staggered schemes is that they allow for subcycling, which can greatly enhance the efficiency of the coupling process. In many practical applications, the time scales relevant to the fluid dynamics and structural dynamics are vastly different. It is often the case, that the fluid dynamics requires much smaller time steps than the structural dynamics. Thus, it is possible to advance the flow for several time steps within step (2) before proceeding to step (3). This reduces the total number of coupled time steps required to perform the simulation, which in turn reduces the total number of operations required to complete an entire simulation. This approach is depicted in Fig. 7b, where n refers to the number of subcycling steps taken.

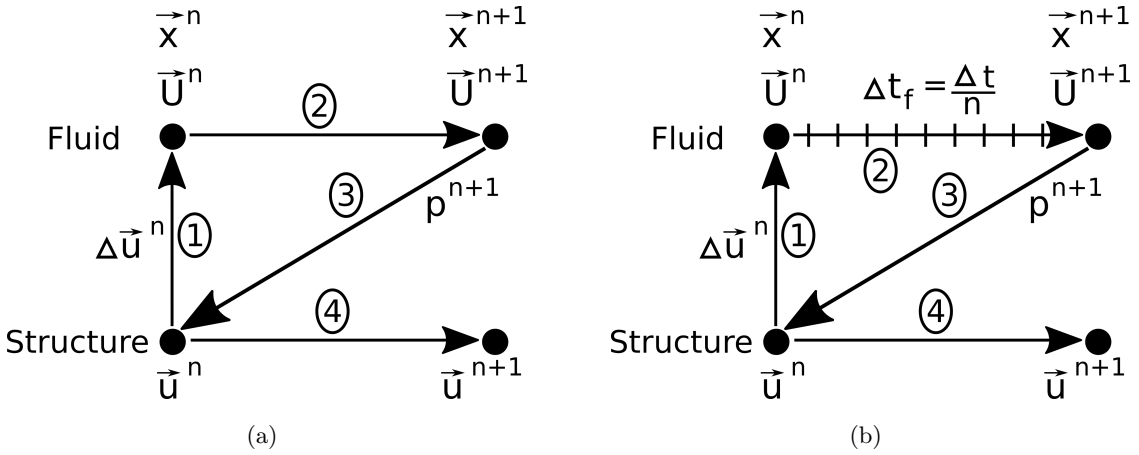



Figure 7: Diagrams depicting the Conventional Serial Staggered (CSS) method shown in (a) and CSS method using subcycling in (b).

The GSS algorithms provide schemes that have been proven to be of $2^{nd}$ order accuracy. One form of the schemes were initially proposed by Farhat and Lesoinne [33] and Lesoinne and Farhat [70], and referred to as the Improved Serial Staggered method (ISS). The scheme was generalized in Refs. [40, 35], and summarized as follows for application in the present body of work.

The first form of the GSS method that is considered relies on the use of a predicted structural displacement. The following steps describe the procedure.

1. update the fluid dynamic mesh using the following prediction:
   $x^{n+1,p} = x^{n,p} + \overline{M}\Delta u^{n+1,p}$
2. use this geometry configuration to update the fluid state from time $t^n$ to time $t^{n+1}$
3. apply the fluid dynamic pressure loading ($p^{n+1}$) as nodal forces ($f^{n+1}$) in the structural solver
4. solve the structures system from time $t^n$ to time $t^{n+1}$

It is clear that the only difference in the algorithm comes from step (1), where the mesh coordinates are predicted as a function of current and previous structural states. As in the CSS method, it is assumed that $\overline{M} = 1$ for applications where the CFD mesh is identical to the FEA mesh. The change in predicted displacement is

$$\Delta u^{n+1,p} = u^{n+1,p} - u^{n,p} \tag{13}$$

and $u^{n+1,p}$ is computed as

$$u^{n+1,p} = u^{n,p} + \alpha_o \Delta t \dot{u}^n + \alpha_1 \Delta t \left(\dot{u}^n - \dot{u}^{n-1}\right) \tag{14}$$

where $\alpha_o = 1$ and $\alpha_1 = \frac{1}{2}$. This algorithm is shown in Fig. 8a, and is referred to as GSS1 within the remainder of this document.

A second form of the GSS scheme is also studied, and will be referred to as GSS2 throughout this document. A key difference between this scheme and the first GSS scheme

is that the physical times are offset between the fluid dynamics and structural dynamics by a half time step. The procedure is outlined as follows.

0. If first time step, initialize fluid time to $t^{-1/2} = -\frac{1}{2}\Delta t$ and the mesh to $x^{-1/2} = x_o - \frac{1}{2}\Delta t \dot{u}_o$

1. update the fluid dynamic mesh using the following prediction: $x^{n+1/2,p} = x^{n-1/2,p} + \overline{M}\Delta u^{n+1/2,p}$

2. use this geometry configuration to update the fluid state from time $t^{n-1/2}$ to time $t^{n+1/2}$

3. apply the fluid dynamic pressure loading ($p^{n+1/2}$) as nodal forces ($f^{n+1/2}$) in the structural solver

4. solve the structures system from time $t^{n-1/2}$ to time $t^{n+1/2}$

It is clear from the above formulation that both solvers operate at the half time step, advancing from $t^{n-1/2}$ to $t^{n+1/2}$. However, the temporal offset between solvers enters through the evaluation of the predicted mesh state in step (1):

$$u^{n+1/2,p} = u^n + \alpha_o \Delta t \dot{u}^n + \alpha_1 \Delta t \left(\dot{u}^n - \dot{u}^{n-1}\right) \tag{15}$$

where $\alpha_o = \frac{1}{2}$ and $\alpha_1 = \frac{1}{8}$. The structural displacement at time $t^n$ is determined based on a central difference about $t^{n-1/2}$: $u^n = 2u^{n-1/2} - u^{n-1}$.

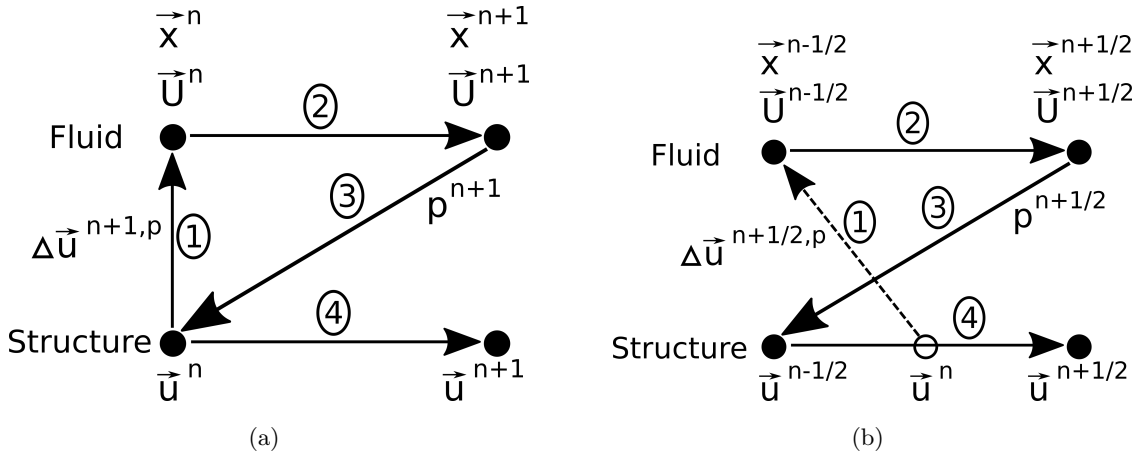



Figure 8: Diagrams depicting two partitioned coupling schemes, with the GSS method 1 shown in (a) and the GSS method 2 shown in (b).

### *2.6 Fluid-Structure Interface*

In partitioned fluid-structure interaction problems, the surface conditions of each solver must be transferred between each other, and applied as boundary conditions. For the fluid dynamics, the surface pressure field and shear stress must be used as a boundary condition to load the structure. In the opposite direction, the displacements and velocities computed in the structural solver must be applied in the surface boundary conditions of the fluid solver. Different approaches must be taken depending on topological differences in the meshes at the interface. Additionally, the conservation of forces, moments, and virtual work are important consideration when choosing an approach.

#### 2.6.1 CFD Data Interpolation to Input Geometry

Due to the immersed boundary method approach used in the present unstructured Cartesian grid solver, a method is required to obtain the surface properties on the input geometry mesh. Knowledge of the fluid properties is a prerequisite for transferring surface pressure data to nodal force data on the structural mesh. Two approaches can be considered for obtaining the surface properties. The first approach is to interpolate the volume flowfield data within the Cartesian mesh directly to the input geometry surface mesh. This approach is straight forward in that a small collection of computational volume cells within close

proximity to the interpolation point are used. One downside to this approach is that, in general, the majority of flow cell data is located exterior to the geometry mesh. There will exist regions where a computational cell center is actually located on the interior side of the surface. However, interpolations using this approach result in largely biased interpolation clouds that do not sufficiently bound the interpolation point. Additionally, interpolations will in general, not accurately enforce surface boundary conditions.

It is for these reasons that a second approach is taken when determining the surface properties on the input geometry mesh. Flow properties are computed using a similar approach to the way the surface boundary conditions are enforced in the flow solver. Details regarding surface boundary conditions are discussed in a later section, but the necessary details for setting the surface flow properties are discussed here.

The fluid state at the wall is extrapolated from a known value within the flow using Eq. 22, where the reference point is determined using Eq. 21. Since the point of interpolation is located on the surface, the value for $\delta_g$ is equal to zero in all cases. Additionally, $\delta_R$ is set to a distance that is equal to the local computational cell diagonal length. Applying these equations to the primitive variables results in the following enforcement at the surface: 1) the velocity condition of impermeability is maintained, 2) the tangential velocity component of slip/no-slip is maintained, and 3) the surface normal pressure gradient vanishes.

### 2.6.2 Similar Meshes at the Interface

The most straight forward situation regarding mesh interfaces occurs when the meshes are identical at the interface. When feasible, the construction of identical meshes can provide a very efficient and accurate mapping and transfer of data. If this is the case, the quantities of interest can be directly copied to the corresponding mesh locations in the matching mesh if the data storage is identical (i.e. node centered vs. cell/element centered data). In the present work, meshes have been designed to adhere to this constraint for 1-D and 2-D problems. Due to the simplicity in designing the structural meshes for these cases, it is highly advantageous from a mesh mapping and data transfer standpoint.

In 1-D simulations the structural mesh elements are made up of only a single node.

Thus, the pressure is applied over an element of unit area, resulting in an equivalence between pressure and force applied to the node. In 2-D simulations, the nodal force must be calculated from the element-based pressure field. Since the meshes are identical, the pressure is directly copied to the element centroids. The actual structural mesh is defined by the element nodes, while a dual mesh can be considered that is made up of the mesh element centroids. The nodal force on the structural nodes is computed by applying the pressure over the dual elements bounded by the centroids of the structural mesh.

### 2.6.3 Dissimilar Meshes at the Interface

In general, computational meshes that are used for each of the analyses will be different. Differences often exist as a result of mesh resolution requirements, especially since regions where large flow gradients exist, don't necessarily correspond to regions of large stresses or strains in the structural domain. Additionally, the topology of the two meshes is often different. In the traditional finite volume and finite difference CFD methods, meshes are constructed around airtight geometries that have an identifiable interior and exterior. However, in the structural analysis of plates, shells, and membranes, there is no requirement for a geometry to form a closed body.

In the present work, dissimilar meshes are used in the 3-D simulations. The analysis of infinitely thin elements in a CFD tool requires a modification that adds an artificial thickness to these regions of the structure. This is exemplified in Fig. 9, where a comparison is made between the symmetry planes of a CFD surface mesh and an FEA surface mesh. The FEA centerline mesh consists of only the portions of the model that undergo deformation or rigid body motion. Thus, the aeroshell and sting are non-existent. The CFD surface centerline mesh demonstrates the use of all components of the model (aeroshell, tension shell, torus, and sting). Note that a finite thickness has been added to the tension shell. It is this subtle difference in the meshes that requires special treatment with respect to the data transfer process.

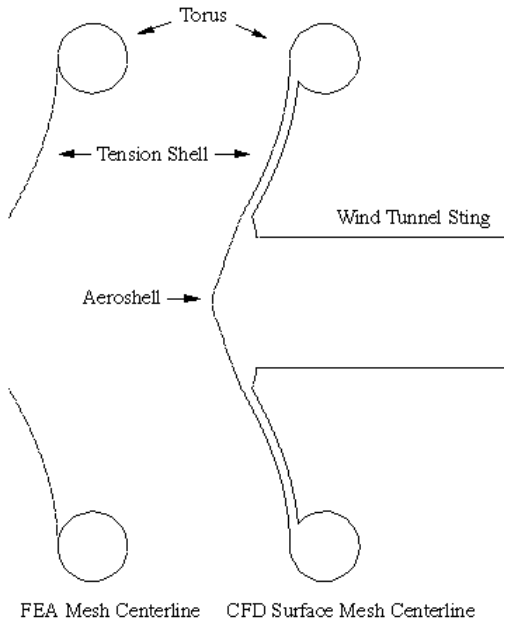



Figure 9: Comparison between an FEA surface mesh and a CFD surface mesh, taken as a slice along the symmetry plane in order to highlight the topological differences. This is an example of a tension cone configuration.

### 2.6.4 Discrete Data Transfer Between Dissimilar Meshes (DDTBDM)

A vast body of data transfer techniques exist in the literature, with emphasis on the consistent and conservative nature of each approach. Smith et al. [103, 102] provides a review of several approaches for aeroelastic applications. Methods include Infinite-Plate Splines (IPS), Multiquadric-Biharmonic (MQ), Thin-Plate Splines (TPS), and Nonuniform B-Splines (NUBS). It was found that the MQ and TPS methods provided the most accurate approach across the tested configurations.

More recently, a software tool was developed at NASA Langley that has been shown to be robust and accurate for performing data transfer for 3-D geometries. The Discrete Data Transfer Between Dissimilar Meshes (DDTBDM) software [96] is capable of transferring data between structured or unstructured meshes that may have topological differences. Quantities can be interpolated between meshes, and when forces are required from a pressure distribution, force injection is possible (conversion from scalar pressure to vector forces). The algorithm ensures force and moment conservation by requiring the total force and moment on each mesh to match after interpolation and force injection.

For these reasons, and the fact that the software was used in similar research [107], the data transfer between NASCART-GT and LS-DYNA is accomplished through the use of DDTBDM.

## *2.7 CFD Solver*

The fluid dynamics is computed using the in-house CFD tool, NASCART-GT. NASCART-GT [74, 73, 69, 10] is a solution adaptive, Cartesian grid-based Navier-Stokes solver that has been in development over the last 15 years. The code was written primarily in FORTRAN 77 with partial upgrades to Fortran 90/95 in the most recent years, following programming styles similar to much of the code developed in the aerospace scientific community. Over the years, NASCART-GT has proven itself with robust grid generation and a wide range of CFD solution capabilities. The solver has been used to compute flows ranging from the subsonic to the hypersonic Mach regimes. Physics models include turbulence and the effects of thermochemical nonequilibrium. Simulations with moving bodies were analyzed, as well as loosely coupled, static FSI problems. Overset gridding capabilities have been demonstrated in the past with applications to moving body problems. Despite the vast capabilities of the software, the ability to efficiently implement new features was becoming more difficult due to a lack of code modularity. As the code developed over the years, more and more features were continuously added. As the code base continued to grow, it was hitting a point where the addition of new complex features was becoming more difficult. Additionally, significant enhancements to memory and parallel efficiency were impractical with the existing framework. Despite some of these roadblocks, the computed physics demonstrated the capabilities of a robust CFD software tool. Several on-going projects presented challenges that provided the impetus to undergo a major upgrade to the software. A few of these included the further development of thermochemical nonequilibrium models and turbulence modeling. Normal Ray Refinement (NRR) [95, 94, 117, 5, 10] was under development as a novel approach to improve the computational efficiency for viscous flows. This approach required significant changes to grid generation and the numerical formulation. Finally, the development of the FSI framework demonstrated a strong need to consider a

software upgrade.

### 2.7.1 Motivation for New Software Development

For all the reasons listed above, there was significant motivation for re-designing the software. Before embarking on this task, a set of goals was established in order to guide the software design process.

1. Performance: The software must be developed using the fastest algorithms and most optimized code whenever possible, though minor losses in performance will be realized in order to keep the code highly modular.

2. Scalability: The software must be written in a way that allows efficient parallelization.

3. Maintainability: This is very important from a code development standpoint. The software must be written in a modular fashion and the inner workings of each module must be properly encapsulated. This leads to well organized code, and increases the rate of future development. Additionally, the code must be written such that all new capabilities will work seamlessly with new or existing physics models.

4. External libraries: The software must use robust and well-tested external libraries whenever possible. This decreases code maintenance requirements and provides access to pre-packaged specialized high performance algorithms.

The first major decision that had to be made was regarding the choice of programming language. Fortran has been used extensively within the scientific community for decades, primarily due to its speed advantages over unoptimized C/C++ and the massive library of legacy codes written in it. Like any software, the speed of the code hinges on the proper and optimal implementations made by the developers. With that said, there are certainly many instances where a C++ code could be as fast or faster than a Fortran code. Thus, the primary factor in making the decision should not fall solely on the speed and optimization potential of the code. When developing software in an academic environment there are several challenges, primarily the turnover rate of students. The ability for new students

to quickly learn how to develop within a new environment is critical. Thus, the concepts of modularity and encapsulation become necessary in the design of a well-written code. These concepts allow code to be written such that each component is, to varying degrees, isolated from the rest of the code. This yields significant advantages when developers only need to concern themselves with small portions of a large software suite and only need to know how to interact with portions of code they are not developing. C++ certainly has its strengths in object-oriented programming, which provides a framework for writing modular code. When looking at the overall picture, it is easier to develop and maintain well-written, optimized code in such an environment.

Code development has focused on the use of the C++ Standard Library and utilizes the linear algebra library, Eigen [47]. Eigen is used extensively throughout the code in order to facilitate the representation of vectors and coordinates through its Matrix class, and to perform linear algebra operations. Eigen implements a number of advanced algorithms and vectorization optimizations in order to provide highly performant and modular capabilities that do not need to be maintained by NASCART-GT developers. Eigen's geometry capability (such as the Eigen::AlignedBox) is used extensively for the efficient representation and computation of cell and surface topologies. For example, each computational cell's topology can be expressed as a Eigen::AlignedBox with which it is computationally efficient to determine if a point is contained within the cell. Similarly, the C++ Standard Library provides a number of algorithms such as sorting, finding min/max elements, and finding values, and data structures like resizable vectors, linked lists, efficient Fortran-like arrays, which are written in a fashion such that they will work for any contained data type. These capabilities decrease the learning curve for new students who are already familiar with C++, rather than requiring them to learn new programming patterns specific to the code, or worse, having to re-implement the algorithms/data structures for their specific use case. Additionally, these algorithms do not have to be optimized by NASCART-GT developers - the C++ Standard Library implementation is provided with the compiler, and thus compiler vendors are free to provide highly optimized versions of these algorithms, which will operate within NASCART-GT seamlessly.

### 2.7.2 Code Design

The following discussion highlights minor coding related items, and style choices made regarding the development of the code. These are presented here to provide a documented resource for the reasoning behind some of the programming choices made throughout development. At the time of writing this document, these are considered best practices, but by no means should be taken as hardened rules. Updates should be continually made as newer concepts and best practice methodologies are understood. The type "double" has been selected as the default floating point number type due to its higher precision compared to float, while the memory cost has been deemed manageable given the low cost of memory today. The standard template library (STL) is heavily used, as is the external library Eigen. It was decided that all instances of STL functions must be employed by explicitly preceding the function with the std namespace.

One important feature of the entire implementation is that the code is compiled differently depending on the number of dimensions. By specifying the dimension at compile time, significant portions of code can be excluded in the executable that is built. This also leads to the development of highly optimized implementations based on the number of dimensions. Specifying the number of dimensions at compile time also allows the compiler to perform greater optimizations since more information is known.

A header file (Utils.hpp) has been created for use by all other files in the code base that require the core functionality. This is where common includes are set, as well as the problem dimension, useful global variables, enumerations, and functions. Commonly used constants are specified using the C++ constexpr type specifier in order to facilitate compile-time optimizations. C++ enumerations are used to facilitate readability throughout the code. For example, anywhere directionality is required in the code (i.e. face directions) the enumeration, "Direction" is used, with available choices set to "RIGHT", "LEFT", "TOP", "BOTTOM", "FRONT", or "BACK". Another common enumeration used throughout the code is "Axis", which can be specified as "X", "Y", or "Z". This provides a very clear implementation, which specifies intent. For example, if the Y component of a vector, v, must be assigned, it can be done so using v[Y] instead of v[1]. Several other enumerations

are used, which correspond to scheme choices, algorithms, and other common parameters.

### 2.7.3 High Level Structure

The main C++ file contains minimal implementation, and acts as the owner of the entire simulation, executing the process, and cleaning up all resources. The main function creates an object of the Domain class, which orchestrates the entire numerical problem. The Domain obtains the user specified input by creating an object of the InputFile class to read and store all information pertinent to the case. The type of simulation dictates how the solution must proceed, and this information is stored within the Simulation class. Once the Domain class instantiates a Simulation object, it must first initialize the Simulation, and then advance the simulation in time though numerical iterations.

The Simulation class is the base class for any derived implementations of specific physical problems. The Simulation creates the grid and all numerical models that are required for the problem at hand. For the common case of the Navier-Stokes equations, the Navier-Stokes class is used, which contains an object of the Grid class, a TimeIntegration scheme, and a SpatialScheme. Other examples consist of the GeometryMotion, PanelMethod, Newtonian, and FSI classes. The GeometryMotion class consists only of the input physical geometry. Such a simulation does not require a grid or spatial and temporal discretization. In order to make this simulation non-trivial, a body motion should be defined, such as a 6-DoF model with applied loads. Thus, this class presents an approach for validating the motion capabilities within the code. The PanelMethod class implements an approach for calculating the pressure distributions for subsonic incompressible and compressible flows, using surface information only. The Newtonian class implements both the Newtonian and Modified Newtonian methods for computing pressure distributions over geometries in hypersonic flow. The primary distinction between sub-classes of Simulation is that each may require different combinations of time integration schemes, spatial schemes, grids, and input geometries.

The primary simulation type for this research is the NavierStokes and FSI simulation classes. The solution of the Navier-Stokes equations requires three main components: 1.) a

grid for numerical discretization, 2.) a time integration scheme, and 3.) a spatial scheme. Thus, the remainder of the code descriptions will focus around the Navier-Stokes and FSI implementations.

### 2.7.4 Geometry Configuration and Motion

Before discussing the spatial discretization and grid class, a discussion on the input geometry and motion is necessary. The input geometry configuration is stored in an object of the GeometryConfiguration class. The Grid class, which will be discussed next, is the owner of this object. The class has been designed to store multi-body configurations, along with motion. Any number of input geometries are allowed, each coming from a different geometry file. Each geometry component can undergo independent scaling, initial translations, initial rotations, and unique grid parameters can be specified.

The input geometries are stored in objects of the PhysicalGeometry class. Three types of input geometries are allowed, depending on the dimension of the problem. 1-D simulations accept a set of points that can define the boundary points of the problem. An example of this would be a 1-D piston, defined by two points (left edge and right edge of a finite thickness piston). In 2-D, geometries are specified by a list of coordinates that are converted into discrete line segments. In 3-D, the STereoLithography (STL) file format is required, and converts the triangular information into a triangle element data structure.

For each geometry component, any number of motions can be assigned. Allowable motion types are contained within the following classes: PrescribedTranslation, PrescribedRotation, PrescribedPitch, PrescribedPlunge, RigidDynamics, MassSpringSystem, BeamSolver, or LS_DYNA. All of these classes derive from the base class, BodyMotion. Each of the classes simulates motion through a function that implements the specific type of motion, advances local motion properties, and updates all geometry components that are impacted by the motion.

The prescribed motion classes are used for motion types that are known a priori. Currently, there are four types of prescribed motion, though the extension to a prescribed

generalized motion could be easily added, provided a well defined input format for describing such motion. PrescribedTranslation is specified by a velocity vector that indicates the speed and direction of translation. PresribedRotation is used for a geometry component that is undergoing rotation about a specified axis at a constant speed. PrescribedPitch is specified by a pitching axis, center of rotation, and an amplitude and frequency of oscillation. PrescribedPlunge is specified in the same way as the pitching motion, except the amplitude is a translation amplitude, and the pitching axis becomes the translation axis.

The RigidDynamics class is used for six DoF motion subject to the applied aerodynamic loads. The input parameters for this motion include the mass, center of mass, and principal moments of inertia. The input file provides the option for constraining any of the six degrees of freedom (translation or rotation). Initial translational and angular velocities can be specified, along with external body accelerations (i.e. gravity). Details regarding the 6-DoF algorithms can be found in Appendix B.

Two simplified structural models are implemented within NASCART-GT to study the interaction with mass-spring systems and Euler-Bernoulli beams. The MassSpringSystem class solves the $2^{nd}$ order, 1-D mass-spring ODE with an applied forcing function derived from the aerodynamic load. Input parameters include the mass and spring stiffness. The equations can be solved for either dynamic or static equilibrium. The Euler-Bernoulli beam implementation solves the equations of motion for a system constructed from elemental mass and stiffness matrices.

The LS_DYNA class implements the functions necessary for preparing data for output, and reading in deflected data for coupling with LS-DYNA. Several parameters are required for input, including information regarding the LS-DYNA run path, executable locations, input file names, and execution parameters. The simulation functionality includes writing out aerodynamics forces, initiating the execution of LS-DYNA, reading in the deflected geometry, and updating geometry and boundary conditions within NASCART-GT.

### 2.7.5 Grid Generation

The unstructured, Cartesian mesh is automatically generated, and is resolved to a user specified level to capture the immersed geometry. The Grid class is used to manage the entire mesh, including grid generation, solution adaption, and visualization output of mesh points. The grid stores all of its data in an Octree data structure. An Octree is a data construct that serves as a container for what are termed, children cells. The primary advantage of using such a structure is the parent-child relationship that can be used to easily traverse computational cells. This is graphically described in Fig. 10.

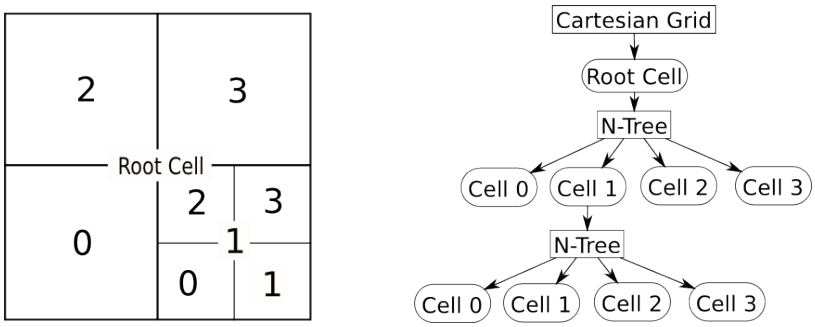



Figure 10: Cell-octree data structure.

In its simplest approach, the grid contains a single cell, called the root cell. The root cell has no parent since it is the highest cell in the data structure. As additional cells are added to the mesh, they are initialized by refining the root cell. Refinement consists of creating a new child octree within the current cell. The octree is a container itself that has connectivity information to its parent cell and it children cells. Upon creation, $2^{Dim}$ children cells are created, where $Dim$ is the dimension of the problem. This process of refinement continues as necessary in order to resolve the input geometry. Now, consider the case where the base of the grid is constructed by using multiple root cells. This setup produces multiple octree data structures that are linked by assigning neighbor cells for each root cell. The purpose for having such a capability is that is provides a means for generating rectangular grids, while still maintaining square cells within the mesh.

The grid generation process is primarily governed by the continual reduction in cell size at the grid/geometry interface. During this initial grid generation process, the actual

intersection points between the physical geometry and the mesh are not necessary since it is only refinement that is important. Thus, the algorithm used for determining whether an intersection exists can be far more optimized than calculating intersection points. Cells are continually sub-divided until user-specified grid generation criteria are met. The main criteria requires a minimum number of computational cells along the longest body dimension. Upon completion of surface refinement, the intersection points are calculated using algorithms specific for each of the three physical geometry types.

All cells that contain intersection points with the physical geometry are called surface cells since they contain a representation of the physical geometry surface. Each surface is treated as a single line segment (2-D) or a plane (3-D) with intersection points and a single unique unit normal vector. This data is stored in an object of the Surface class, where each cell contains a unique pointer to it. The surface boundary conditions are applied by using this information, which implies that the surface boundary conditions can only be as accurate as the surface implementation allows.

At this point during grid generation, the mesh consists of cells that are both external and internal to the geometry configuration. A flood fill algorithm has been implemented to set the state of each cell. Provided that there is a cell which is known to be external (in most cases a corner cell of the mesh), it can be used as a starting point. The algorithm proceeds by setting every neighbor of the starting cell to a boolean "inside" value of false. This process continues by setting the neighbors' neighbors until a surface cell is reached. Surface cells are still treated as external cells, but the flood fill terminates the search along that path since doing so would enter the inside of the geometry.

In most simulations, it is desirable to have a region of cells that extends outward from the surface of the geometry configuration, where the refinement level of the cells are uniform and all assigned to the level of the surface cells. This region is called the uniform surface layer. Having such a region improves the quality of the solution near the surface by reducing errors in the application of the surface boundary conditions. Additionally, when running viscous simulations, there is the need to have regions of highly refined cells near the surface in order to capture the strong gradients in the boundary layer.

Finally, the grid must contain information regarding the state of the flow within every cell. This is done by storing information within the Solution class. This class contains the conservative state vector and the time step required based on the local fluid state.

### 2.7.6 Time Integration

The TimeIntegration class provides the framework for creating child classes that implement different types of time schemes. Each time integration scheme follows a different procedure, but in general, requires the evaluation of the right hand side of Eqn. (16) at one or more points in time. Several explicit time schemes are implemented, including the $1^{st}$ order accurate explicit Euler scheme, the $2^{nd}$ order accurate Hancock Predictor-Corrector scheme, and six Runge-Kutta schemes up to $4^{th}$ order accuracy. The functionality for the implicit scheme, LUSSOR, also exists, which provides a $1^{st}$ order accurate fully implicit option and a $2^{nd}$ order accurate semi-implicit option.

$$\frac{\partial}{\partial t}\left[\overrightarrow{U}(\overrightarrow{x},t)\right] = F(\overrightarrow{x},t) \tag{16}$$

### 2.7.7 Spatial Scheme

The SpatialScheme class contains a combination of all numerical schemes that are required to solve the right hand side of Eqn. (16). The equation can be broken down into terms that can be classified by their numerical properties. For example, the integral form of the Navier-Stokes equations can be written as

$$\frac{\partial}{\partial t}\int_V \overrightarrow{U}\,dV = \oint_A \overline{H}\cdot d\overrightarrow{A} + \overrightarrow{S} \tag{17}$$

where the flux tensor is given by

$$\overline{H} = [\overrightarrow{E}, \overrightarrow{F}, \overrightarrow{G}] \tag{18}$$

and $\overrightarrow{E}$, $\overrightarrow{F}$, and $\overrightarrow{G}$ are the flux terms in each coordinate direction and can be split into convective (inviscid) and dissipative (viscous) components. The source term(s), $\overrightarrow{S}$, can

represent any physical source(s) modeled by the governing equations. The closed integral is over the surface of the control volume, $A$, and $V$ is the computational cell volume. Discretizing in space over a computational cell yields

$$\frac{\partial}{\partial t}\overrightarrow{U} = \frac{1}{V}\left[\sum_{faces} \overline{H}_{conv} \cdot \overrightarrow{A} + \sum_{faces} \overline{H}_{visc} \cdot \overrightarrow{A} + \overrightarrow{S}\right] \tag{19}$$

Finally, Eqn. (19) can be conceptually viewed as

$$\frac{\partial}{\partial t}\overrightarrow{U} = [ConvectiveTerms] + [ViscousTerms] + [SourceTerms] \tag{20}$$

where each of these components vary for different sets of governing equations. A vector of these schemes is created inside the SpatialScheme class, and solutions are computed by looping over each scheme's implementation.

The Scheme class is the base class for every scheme that is implemented in order to solve the right hand side of Eqn. (16). The inheritance diagram of the Scheme class is shown in Fig. 11. The convective, viscous, and source term components are shown, and inherit from the base Scheme class. Under each category, there are inherited classes that implement the specific schemes. This design provides an efficient means for code development in terms of adding new schemes, without impacting any other regions within the code.

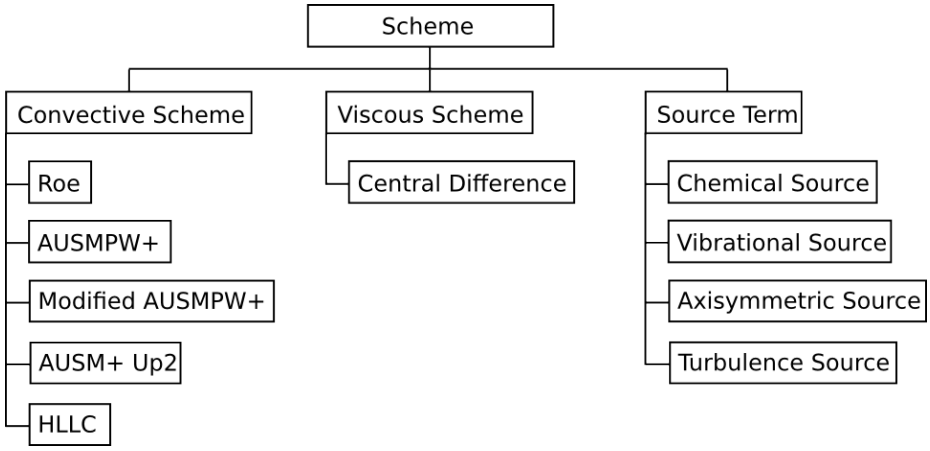


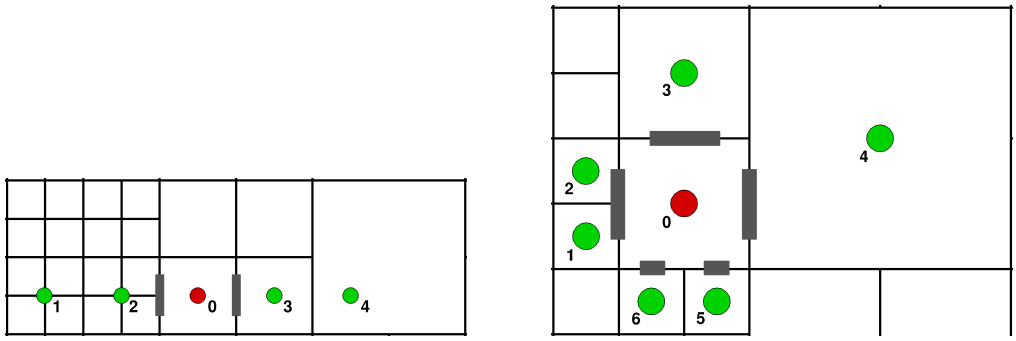

Figure 11: Inheritance diagram of the Scheme class.

### 2.7.8 Convective Flux Schemes

The implemented convective flux schemes are all face-based approaches, that solve Riemann problems at each face, and sum contributions over all faces. The ConvectiveScheme contains the functionality to loop over faces, and appropriately add the contributions. The actual flux calculations are computed using the derived class for each specific convective flux scheme. Several schemes are implemented, including AUSM, AUSMPW+, modified AUSMPW+, AUSM+up2, Roe's scheme, and HLLC. Modifications to AUMSPW+, which take into account the ALE formulation can be found in Appendix A.

The numerical stencil is a vital component in determining the fluxes that are used for the convective and viscous schemes. Two approaches exist within NASCART-GT for computing the fluxes. The first, is based on a local structured grid approach, where neighbor information is determined based on the cell size of the cell being updated. An example of this is shown in Fig. 12a, where the red cell is the cell being updated, and the green cells are the cells required to updated the state. For neighbor cells that are larger than the current cell, the state contained at the cell center of the larger cell is used in the location identified as "4". For the situation where the neighbor cells are smaller than the current cells, the smaller children cells are averaged to determine the state value at locations "1" and "2".

The second approach that is used is referred to as the unstructured approach, where only face neighbor data is required to compute the numerical fluxes, as seen in Fig. 12b. In this method, state variable gradients are reconstructed within the cell using a weighted least squares approach, and used to extrapolate to each face. The majority of analysis in the present work is performed using the first approach, as it was the first method implemented. The second method has been implemented recently by Dement and Ruffin [29] at the time of writing this document. Any simulations using the second method are explicitly identified where necessary, and reasons for choosing this method include improved accuracy and computational time.




(a) (b)

Figure 12: Stencil construction for (a) structured approach and (b) unstructured approach for extrapolating the state to the faces of cell # 0.

### 2.7.9 Viscous Flux Scheme

The ViscousScheme class is a pure virtual base class, and the child class, NSCentralDifference2, implements a $2^{nd}$ order accurate central difference flux scheme. In line with the convective flux schemes, the contributions from each face are summed, and added to the total cell residual.

### 2.7.10 Source Terms

Source terms are treated with a slightly different approach, since their mechanism is inherently different. All source terms that are implemented are not face-based, but rather based on cell-centered data. Currently, there are several source terms including axisymmetric terms, chemical rate productions, vibrational energy production, and a turbulence production source term. These terms are added to the overall residual calculation.

### 2.7.11 Boundary Conditions

The boundary conditions are implemented as a heirarchical class, where the BoundaryCondition class serves as a purely virtual interface. All boundary condition calculations are contained within derived classes that override the virtual interface. These conditions include the surface boundary conditions, Riemann invariants, symmetry plane conditions, superonic outlet, and freestream.

#### *2.7.11.1 Surface Boundary Conditions*

The surface boundary conditions are applied using the immersed boundary method (IBM). Details regarding the original NASCART-GT implementation can be found in references [67] and [99]. The methodology is further detailed here, with emphasis on the effects of non-stationary surfaces.

An example numerical stencil is shown in Fig. 13, where the surface passes through three of the computational cells. The stencil is used to update the state in cell #7, and is characteristic of a $2^{nd}$ order accurate extrapolation. Additionally, corner cells are included, and required for either the calculation of viscous gradients or are used in the pressure weighting calculation in the AUSMPW+ inviscid flux scheme. All computational cells except for #13 are treated as flow cells, in that their states are computed using flux integration. Cells #10, #11, and #12 are identified as surface cells, but no special treatment is applied to them. However, cell #13 is classified as a ghost cell since it is completely inside the body. Thus, the objective of the surface boundary condition is to appropriately set the state at the ghost cell center.

The approach to setting the state at the center of cell #13 is to extrapolate data from inside the flow domain. Two distances are required for this calculation. First, is the distance ($\delta_G$) from the ghost cell center to the nearest surface, along the vector defined by the normal of the nearest surface cell. The second distance ($\delta_R$) determines where the reference point is located for use in extrapolation. Two approaches are implemented for determining this distance. The first is to set $\delta_R = \delta_G$, which is the primary implementation used for setting the state during flux integration. An alternate approach is to set $\delta_R = CellDiagonal$, and is often used when setting the surface state for visualization output.

Once $\delta_G$ and $\delta_R$ are determined, the reference point can be determined as:

$$\mathbf{R} = \mathbf{x}_{13} + (\delta_G + \delta_R)\mathbf{n} \tag{21}$$

For Dirichlet boundary conditions, the ghost cell state is then obtained through an extrapolation, which uses an interpolated state at the reference point, and the known state

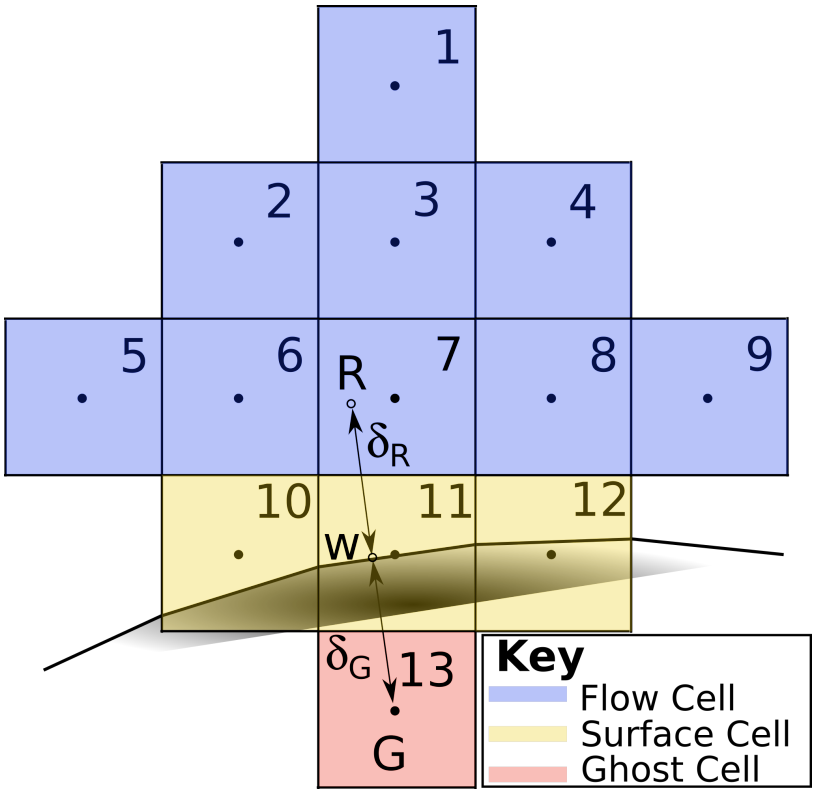



Figure 13: Generic numerical stencil used near surface.

at the wall. Dirichlet boundary conditions are used for setting the no-slip velocity and isothermal temperature. Equation 22 represents a generic boundary condition application of this type, for setting a state, $\phi$.

$$\phi_{13} = \phi_w + (\frac{\delta_G}{\delta_R})(\phi_w - \phi_R) \tag{22}$$

where $\phi_w$ is the prescribed quantity at the surface. Note, that for the no-slip velocity boundary condition, $\phi_w$ becomes the velocity of the wall, which can be non-zero for moving body simulations.

For the slip velocity boundary condition, the wall normal component of velocity must match the surface normal velocity of the wall. The tangential velocity component is unaffected by the existence of a moving surface.

For Neumann boundary conditions, the ghost cell state is obtained by enforcing the gradient to be zero at the wall.

$$\frac{\partial \phi_{w,14}}{\partial n} = 0 \tag{23}$$

This boundary condition is specifically used for the adiabatic temperature condition, and for setting the pressure, assuming a zero pressure gradient.

The implementation of the surface boundary conditions is the most computationally expensive operation of the boundary condition options. The class heavily interacts with the Cell and Grid class in order to determine nearby surface information that is required for setting the extrapolation distances. Additionally, interpolation to the reference point is required, which significantly increases computational cost. The actual extrapolation calculation, where Eq. 22 is used, requires little effort.

When simulating moving-body cases, there is a special situation that occurs when cells emerge from inside the body. Two approaches have been utilized. If the emerging cell is a surface cell (which is typically the case for small movements over a single time step), then the state of the cell is set by applying the surface boundary condition, which provides a physically appropriate value for the state. However, in situations where the geometry moves many cells over a single time step, it is likely that the emerging cells do not become surface cells, and can in fact be quite far away from the body. For this situation, the state is interpolated from the nearest group of usable cell states.

#### *2.7.11.2 Riemann Invariants*

Riemann invariants are typically used to set the sates when using farfield boundary conditions, and can be applied for subsonic and supersonic flow conditions. These conditions use the fact that the Riemann invariant state remains constant along characteristic lines in the flow.

The implementation of this approach requires the states that are interior and exterior to the domain. The interior states are obtained through the use of a reference cell inside the domain. The exterior state is assumed to be the freestream flow conditions.

The characteristics for a perfect gas are listed in Table 1. For supersonic flows, the Riemann invariants are taken from the exterior for inflow, and are taken from the interior

for outflow. For subsonic inflow, $R_1$ through $R_4$ are taken from the exterior, while $R_5$ is taken from the interior. For subsonic outflows, $R_1$ through $R_4$ are taken from the interior, while $R_5$ is taken from the exterior.

Table 1: Riemman invariants for a perfect gas.

| X-Direction | Y-Direction | Z-Direction |
|---|---|---|
| $R_1 = \frac{p}{\rho^\gamma}$ | $R_1 = \frac{p}{\rho^\gamma}$ | $R_1 = \frac{p}{\rho^\gamma}$ |
| $R_2 = v$ | $R_2 = w$ | $R_2 = u$ |
| $R_3 = w$ | $R_3 = u$ | $R_3 = v$ |
| $R_4 = u + \frac{2a}{\gamma - 1}$ | $R_4 = v + \frac{2a}{\gamma - 1}$ | $R_4 = w + \frac{2a}{\gamma - 1}$ |
| $R_5 = u - \frac{2a}{\gamma - 1}$ | $R_5 = v - \frac{2a}{\gamma - 1}$ | $R_5 = w - \frac{2a}{\gamma - 1}$ |

#### *2.7.11.3 Symmetry Plane*

The symmetry plane condition sets the state such that all flow properties are matched to the reference point, except for the normal component of velocity. Such conditions are often used along a plane dividing a symmetric flow around a symmetric body. The condition is also used when simulating an inviscid wall at the domain edge. In fact, the implementation of this class is a specialized version of the surface boundary condition, where the reference point is known, the wall normal is pre-defined, and the velocity condition is set to slip wall.

#### *2.7.11.4 Supersonic Outlet*

The supersonic outlet boundary conditions can be used when the flow at the exit is known to be supersonic. Since information does not propagate upstream in supersonic flow, the condition downstream is only a function of the upstream state (as opposed to the subsonic Riemann invariant approach). Information still propagates along the characteristic directions, but can be approximated by setting the exterior state to be equal to the state immediately interior to the domain.

#### *2.7.11.5 Freestream*

The freestream boundary condition is a straightforward implementation that immediately returns the global freestream state of the flow. This is generally only used to apply the

boundary condition at the inlet for supersonic cases. Subsonic cases require Riemann invariants at the inlet.

### 2.7.12 Solution-Based Grid Adaption

Solution-based grid adaption is a feature that is used to ensure that there is sufficient grid resolution throughout the computational domain. For steady-state simulations, the solution converges to a flowfield that is frozen in time. Thus, the purpose of grid adaption is to ensure that the final mesh used to converge a solution is optimal with respect to the final solution. For unsteady problems, where the flowfield's spatial gradients are continuously changing, the purpose of solution adaption is to maintain an optimal mesh for all time.

Grid adaption is implemented as a functionality within the Grid class. The core functionality is based on the refinement strategy described by Tu [108], Marshall [75], Lee [68], Lee [67], and Sekhar [99]. The input parameters that drive this functionality are the iteration numbers at which the process begins and ends, as well as the frequency for which the mesh is updated. The refinement and unrefinement flags are set based on gradients in the flowfield.

The gradients are calculated for each computational cell using

$$\tau_\phi = \sqrt{|\vec{\phi}|^2 V} \tag{24}$$

where $\vec{\phi} = \vec{\omega}$, $\vec{\nabla} \cdot \vec{V}$, $\vec{\nabla} P$, $\vec{\nabla} T$, $\vec{\nabla} M$, $\vec{\nabla} T_v$, or $\vec{\nabla} y_i$. These correspond to vorticity, velocity divergence, pressure gradient, temperature gradient, Mach number gradient, vibrational temperature gradient, and species mass fraction gradient respectively.

The mean gradients are calculated using

$$\sigma_\phi = \frac{1}{N} \sum_{n=1}^{N} \tau_\phi^2 \tag{25}$$

and cells are refined or coarsened based on the following conditions:

$$coarsen : \tau_\phi < \kappa_c \sigma_\phi \tag{26}$$

$$refine : \tau_\phi > \kappa_r \sigma_\phi \tag{27}$$

The values for $\kappa_c$ and $\kappa_r$ are specified as user input parameters for coarsening and refinement respectively.

### 2.7.13 Variable and Mixture Classes

The conservative state vector is stored at every cell center, though many other flow properties must be calculated throughout a large scope of the solver. All functions that are needed to make such computations are done via the Variable class. The class stores the freestream state, along with information about additional models used (i.e. mixture and turbulence models). The functions act as a generalized interface to calculate flow properties that can be derived from the conservative state vector.

The Variable class also contains pointers to the mixture model and turbulence model, if relevant. The Mixture class is a pure virtual class with derived classes for calorically perfect gases (CPG) and chemically reacting perfect gas mixtures. This approach hides the specific implementations necessary for computing thermodynamic quantities behind Mixture functions, so the Variable functions do not need to know the specifics of the mixture model. This further implies that the rest of the code base does not need to be aware of these details either. For example, when calculating flux scheme values, there is no need to alter the expressions based on the type of mixture model used.

## *2.8 FEA Solver*

The structural analysis for FSI simulations can leverage the use of any number of commercially available tools, open-source software, or the development of in-house codes. Rohrschneider [93] investigated several structural analysis tools, including LS-DYNA, ABAQUS, and ANSYS. It was found that LS-DYNA was the best choice for applications such as inflatable aerodynamic decelerators, where the materials are thin, membrane-like fabrics. Tanner [107] expanded upon this research and continued to use LS-DYNA, citing Rohrschneider's assessment of numerical stability and consistent accuracy. That work also examined several

verification cases with applications to inflatable decelerators. Additionally, LS-DYNA has a large user base in the decelerator community. Examples of relevant structural analyses using LS-DYNA can be found in Refs. [24, 109, 110].

For these reasons, LS-DYNA has been chosen for the current research. However, the proposed computational framework shall not be limited to LS-DYNA, and in general, any alternate structural analysis tool could be chosen if it satisfies engineering requirements. It is important that the structural dynamics is modeled with sufficient accuracy, but the primary focus of this research is into the use of the Cartesian CFD framework using the immersed boundary method, and how well it can be used to effectively solve time accurate FSI problems.

LS-DYNA [25] is a nonlinear, finite element solver developed by the Livermore Software Technology Corporation (LSTC). Several element formulations exist within the software package, where the necessary elements for the current work include mass elements, discrete elements, beam elements, and shell elements. For all simulations, the materials are modeled as either linear elastic or isotropic fabrics. When the structural dynamics are relevant, the model is advanced in time using the classic Newmark-Beta method. For steady-state simulations, where the dynamics are not relevant, the equilibrium solver is employed to compute displacements. In order to couple NASCART-GT and LS-DYNA, a means for sending and receiving data is necessary. LS-DYNA has a restart capability that is sufficient for steady-state simulations. However, it was found that using the restart process was insufficient for dynamic simulations where coupling requires data transfer at every time step. For these cases, a user-defined loading feature is utilized. This approach required development within the user-defined libraries of LS-DYNA to properly "pause" LS-DYNA when the structural time step was complete, and to read-in the applied loading before LS-DYNA proceeds with computations. The implementation of this capability is not naturally supported by LS-DYNA at present, and is an important consideration in choosing LS-DYNA for use in time accurate, staggered FSI algorithms.

# CHAPTER III

# STATIONARY & PRESCRIBED MOTION SIMULATIONS

The first step towards evaluating FSI problems is to consider the stationary analysis of relevant geometries and flight conditions as a means of initial code validation. Second, the effects of moving bodies must be explored at the most fundamental level. Prescribed rigid body motions satisfy this requirement, and provide a useful set of cases to validate moving body problem.

In this chapter, numerical simulations are presented, which aim to validate stationary and prescribed moving body cases. Simulations of a rigid tension cone at two different angles of attack serve as a validation of the 3-D, stationary, inviscid capability of the code. The fundamental problem of a supersonic piston moving at constant velocity is heavily discussed to provide validation of the moving boundary conditions, as well as provide a simple problem for gaining insights into the nuances related to the numerics of moving body problems. The complexity of the simulations is then increased by considering the following 2-D flows: a supersonic diamond airfoil in motion and a supersonic cylinder in motion. Moving body simulations are compared to analogous stationary body simulations. Finally, the aerodynamic characteristics of an oscillating airfoil in transonic flow are studied, and results are compared against experimental data and other computational simulations. Details concerning the effects of solution-based grid adaption and the impact of moving bodies on the spatial order of accuracy are also explored.

## *3.1 Rigid Tension Cone*

The first case presents an inviscid analysis of supersonic flow over a rigid tension cone geometry. This serves as a validation of the CFD solver's 3-D steady state inviscid capability. Experimental schlieren imagery and pressure coefficient data [23] are used as a means to validate the computational solution. The experimental data was produced in the NASA Langley Unitary Plan Wind Tunnel (LUPWT). The validations are made for Mach 2.5 flow

at an angle of attack of $0^o$ and $20^o$. The freestream density and pressure are 1.082 x $10^{-1}$ kg/m$^3$ and 4,481.59 Pa, respectively. The $0^o$ angle of attack case was run using $2^{nd}$ order accurate fluxes, while the $20^o$ angle of attack case used $1^{st}$ order accurate extrapolation to the cell faces, and both used the AUSMPW+ inviscid flux scheme. The simulations were advanced in time using a $1^{st}$ order accurate explicit time integration scheme. The grid resolution was set such that the longest body dimension contained a minimum of 512 cells. The simulations were run with a symmetry plane along the $Z = 0$ plane and a grid size of ($X_{min}$,$Y_{min}$,$Z_{min}$) = (-0.3, -0.3, 0.0)m and ($X_{max}$,$Y_{max}$,$Z_{max}$) = (0.3, 0.3, 0.6)m. The boundary conditions surrounding the domain were set to Riemann Invariants except along the symmetry plane. Solution-based grid adaption was employed during the simulation in order to resolve gradients in the flow. Adaption was based on velocity divergence, where the refinement threshold was set to 1.3 and the unrefinement threshold was set to 0.7. Solution adaption was set to occur every 200 iterations, beginning at 800 iterations and ending at 8,000. The simulations ran for a total of 10,000 iterations.

### 3.1.1 Rigid Tension Cone: $0^o$ Angle of Attack

The first validation is for the $0^o$ angle of attack flow. For this configuration and flow conditions, it would be possible to simulate only one quarter of the geometry. However, the entire positive half of the geometry was used for the flow computation. The computational flowfield is compared to the experimental Schlieren imagery. The primary focus of this investigation is to verify that the bow shock locations agree. It can be seen in Fig. 14 that there is fairly good agreement in the overall shock structure, while the standoff distance is slightly over-predicted. There is some disagreement with the shock structure on the aft-side, but this is not a primary concern since an inviscid analysis is not able to capture viscous effects that dominate the structure of the wake.

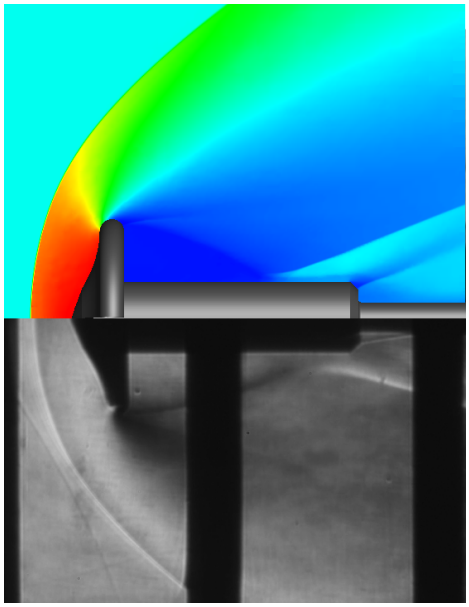


Figure 14: Comparison between the computed density field and the experimental schlieren image for the rigid tension cone at $0^o$ angle of attack.

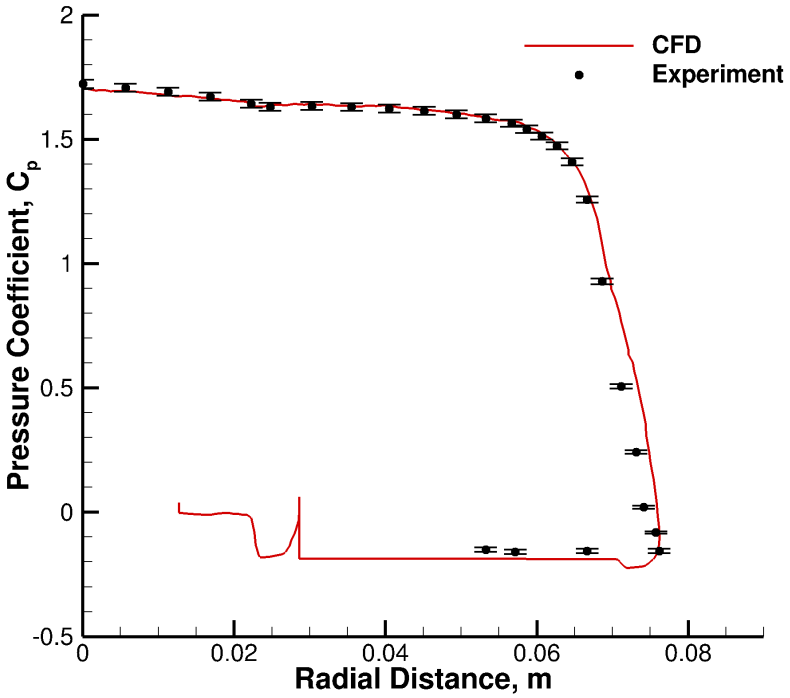
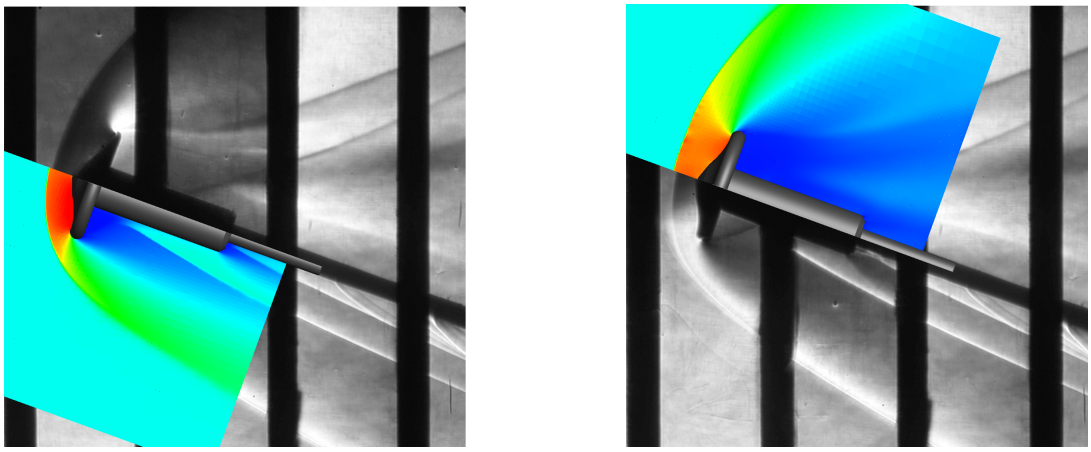



Figure 15: Pressure coefficient along the symmetry plane of the rigid tension cone at $0^o$ angle of attack.

The pressure coefficient distribution is compared against experimental measurements in Fig. 15. The stagnation point pressure matches well, as does the remaining pressure distribution, especially around the expansion at the shoulder.

### 3.1.2 Rigid Tension Cone: $20^o$ Angle of Attack

The next validation takes the same setup as the previous, with a change in flow direction. Figure 16 shows comparisons between the computed density field and schlieren imagery. These comparisons show close agreement with respect to the shock structure in both the forebody region and wake region.



(a) Bottom half of CFD solution.

(b) Top half of CFD solution.

Figure 16: Comparison between the computed density field and the experimental schlieren images for the rigid tension cone at $20^o$ angle of attack.

Figure 17 shows the distribution in pressure coefficient along the symmetry plane. The computational solution compares very well with the experimental data along the windward portion of the surface with an over-prediction occurring as the leeward shoulder is approached.

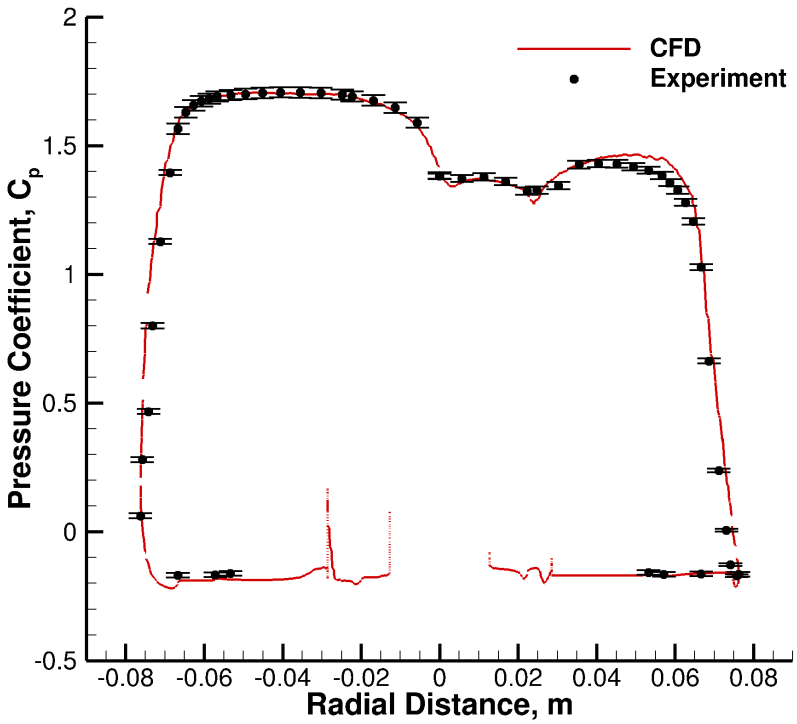



Figure 17: Pressure coefficient along the symmetry plane of the rigid tension cone at $20^o$ angle of attack.

### *3.2 Constant Velocity Piston*

Given the non-stationary nature of the surface interactions in FSI problems, it is necessary to analyze situations with moving boundaries. The first series of analyses presents one dimensional, inviscid simulations of a piston moving at a constant supersonic velocity inside a channel. An analytic solution to this problem exists, in which the flow properties are known as functions of space and time. The solution to such a problem is presented in texts such as John [54] or Anderson [1]. The physical processes taking place in such a simulation include the generation of a moving shock wave ahead of the piston, and a region of expansion waves propagating behind the piston. These two physical phenomena are important in the process of evaluating the numerical properties of the current approach.

### 3.2.1 Fine Mesh Results

#### *3.2.1.1 Stationary Piston*

The first analysis has been setup in a one dimensional domain with a stationary piston of width $2m$, centered about an x location of zero. The boundary conditions at the left and right edges of the domain are set to use the farfield Riemann invariants. The flowfield is initialized to $u = 1,000m/s$, $P = 4 \times 10^5 Pa$, and $\rho = 1kg/m^3$. The freestream velocity corresponds to a Mach number of 1.34 with respect to the speed of sound derived from the initial conditions.

The grid extends from $x_{min} = -30m$ to $x_{max} = 30m$, with a uniform distribution of cells at a refinement level of 10, where $\Delta x \approx 5.86 \times 10^{-2} m$. The inviscid fluxes were computed using the AUSMPW+ scheme with $2^{nd}$ order accurate MUSCL extrapolation. Time integration was performed using the $1^{st}$ order accurate forward Euler scheme, with a constant time step of $\Delta t = 2 \times 10^{-7} s$. It has been observed that for these types of simulations, the contribution of error due to temporal resolution is negligible compared to the spatial error. Thus, the $1^{st}$ order accurate scheme has been used primarily for its computational speed, compared to multi-stage time schemes.

Simulations were run for a flow time of $0.01s$. Figures 18 – 21 show comparisons between the numerical simulations and the analytic solutions. The largest discrepancies can be seen in the density distributions within the immediate vicinity of the piston on both the compression and expansion sides, as shown in Fig. 19. These artifacts are the result of a well known issue that appears in a class of numerical problems, termed the Noh problem [86]. This numerical instability is generally referred to as the numerical overheating problem, where the temperature artificially increases near the wall, resulting in a corresponding decrease in density. The cause of this instability has been largely unsolved. Recent work by Liou [72] attributes the cause of this issue to the pressure flux in the finite volume formulation, additionally citing that it cannot be eliminated through variations in time step, grid spacing, or spatial order of accuracy. Liou proposed a correction that requires the solution to the entropy transport equation instead of the energy equation. Addressing this correction is beyond the scope of the present work, but has been discussed in order

to sufficiently identify this well known numerical issue within the CFD community. The velocity and pressure does not exhibit this numerical instability, as seen in Figs. 20 and 21. The velocity and pressure experience their largest errors at the foot and head of the expansion region, but are nearly indistinguishable from the analytic solution along most of the domain.

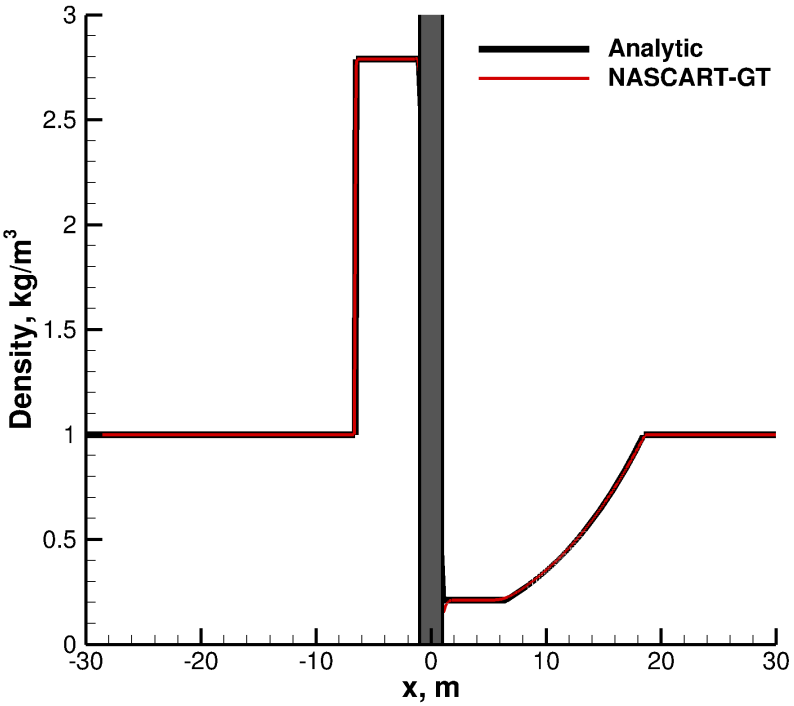
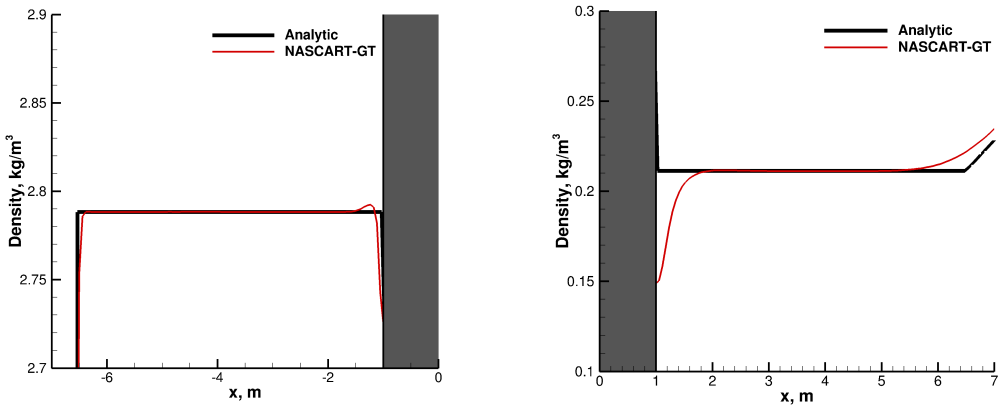



Figure 18: Density distribution for stationary, 1-D piston simulation, where NASCART-GT result is compared against analytic solution.




(a) Compression side.

(b) Expansion side.

Figure 19: Detailed view of the density distribution in the region surrounding either side of the stationary piston.

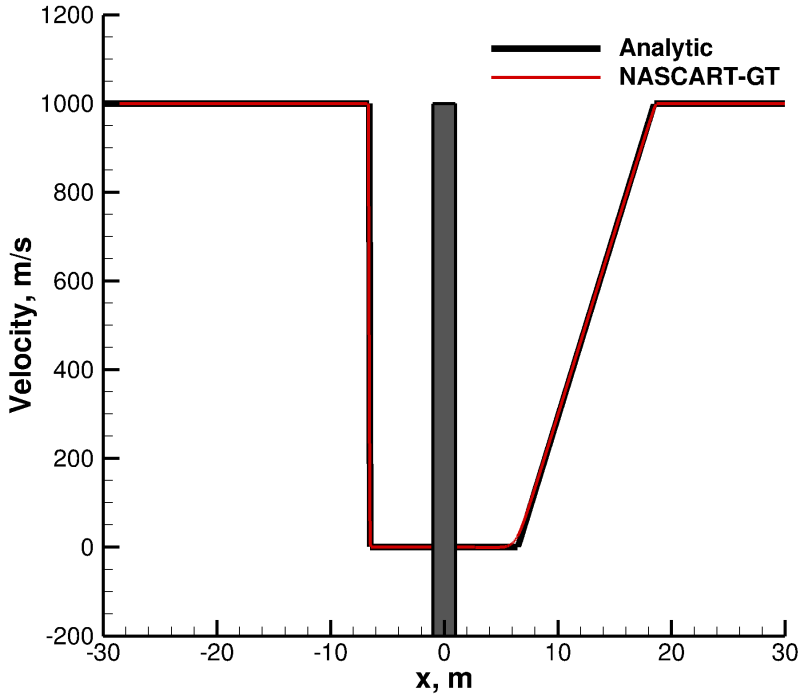



Figure 20: Velocity distribution for stationary, 1-D piston simulation, where NASCART-GT result is compared against analytic solution.

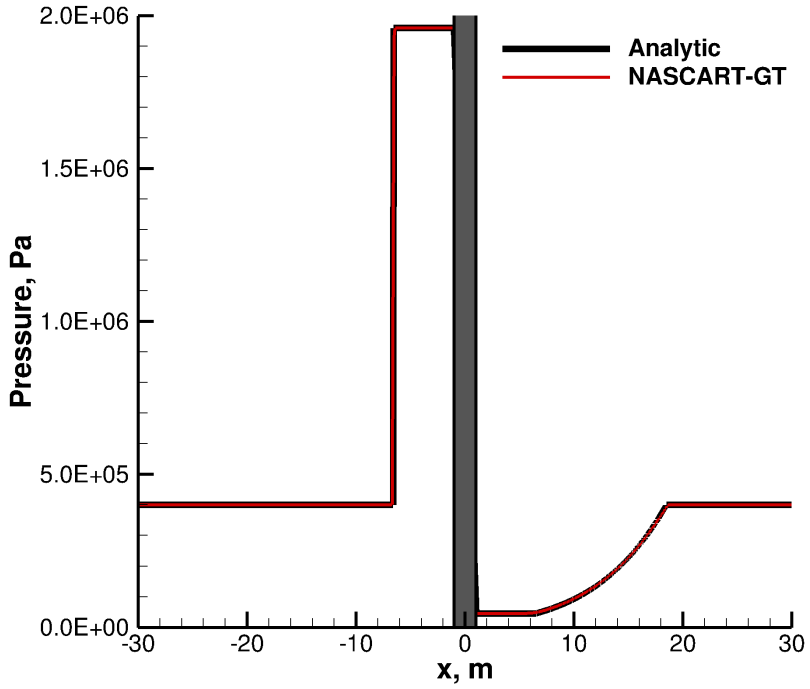



Figure 21: Pressure distribution for stationary, 1-D piston simulation, where NASCART-GT result is compared against analytic solution.

#### *3.2.1.2 Translating Piston*

The translating piston simulation is setup identically to the stationary case. The piston is initially centered about an x location of zero, and travels at a constant, prescribed velocity of $u = -1,000 m/s$, with the negative sign indicating that the piston moves to the left. The flowfield is initialized with zero velocity.

With simulations running for a flow time of $0.01s$, the piston travels a distance of $10m$. Figures 22 – 25 show comparisons between the numerical simulation and the analytic solution. Similar to the stationary results, the largest discrepancies can be seen in the density distributions within the immediate vicinity of the piston on both the compression and expansion sides, as shown in Fig. 23. Compared to the stationary simulation, there is greater numerical dissipation, indicated by the reduction in slope of the gradient. The velocity and pressure does not exhibit the overheating problem, as seen in Figs. 24 and 25.

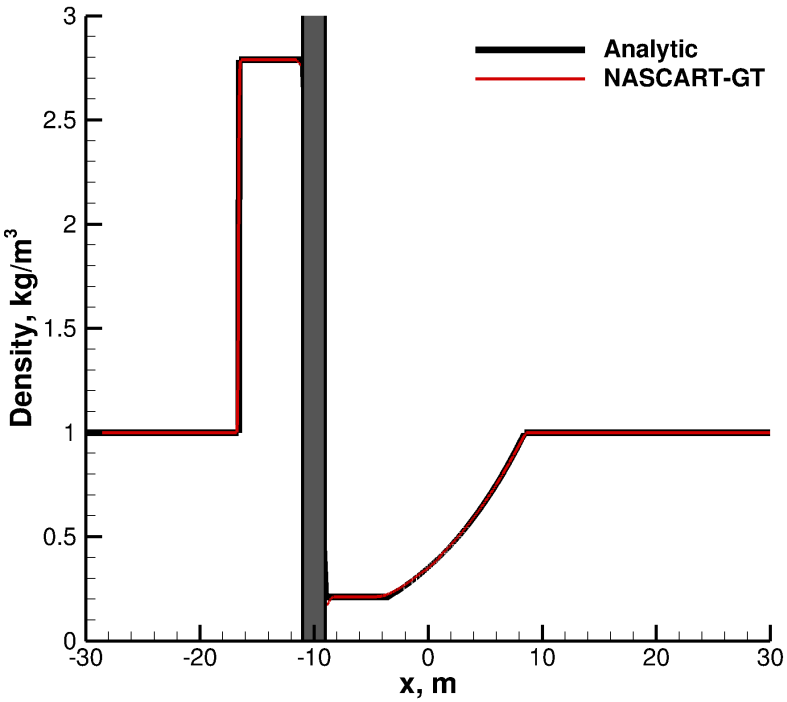
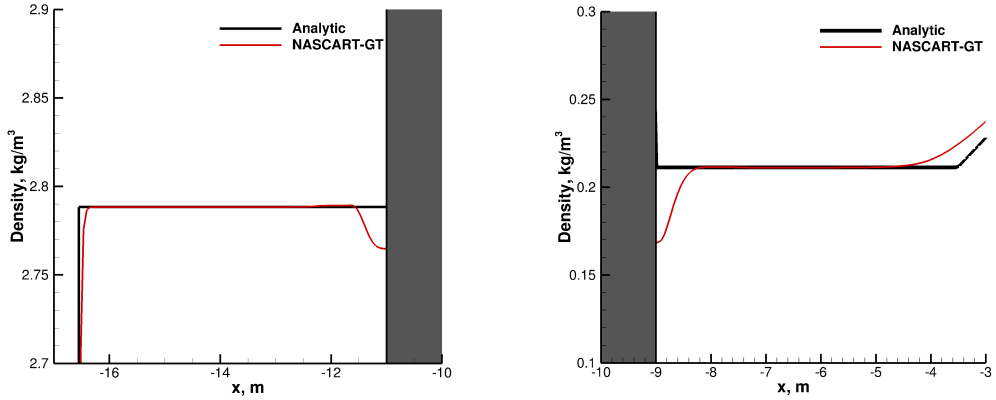



Figure 22: Density distribution for translating, 1-D piston simulation, where NASCART-GT result is compared against analytic solution.




(a) Compression side.

(b) Expansion side.

Figure 23: Detailed view of the density distribution in the region surrounding either side of the piston.

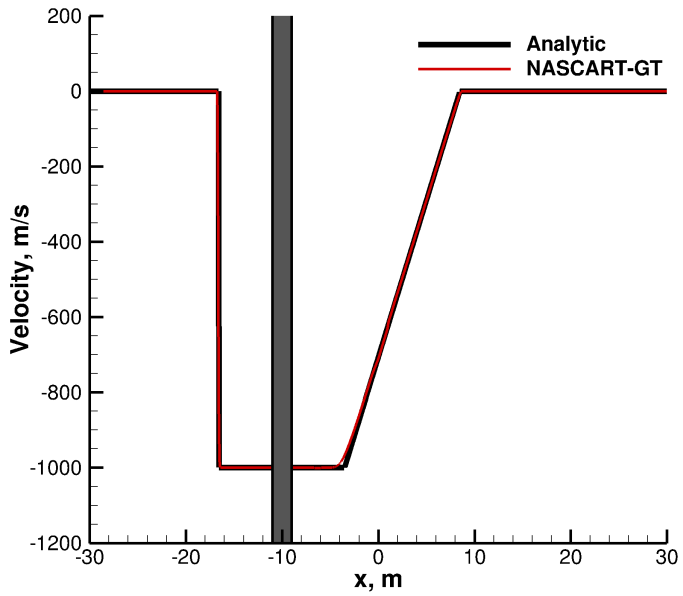



Figure 24: Velocity distribution for translating, 1-D piston simulation, where NASCART-GT result is compared against analytic solution.

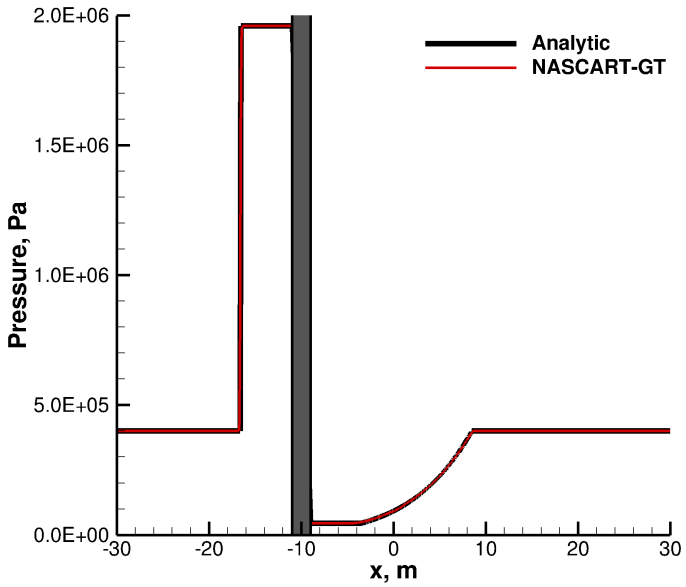



Figure 25: Pressure distribution for translating, 1-D piston simulation, where NASCART-GT result is compared against analytic solution.

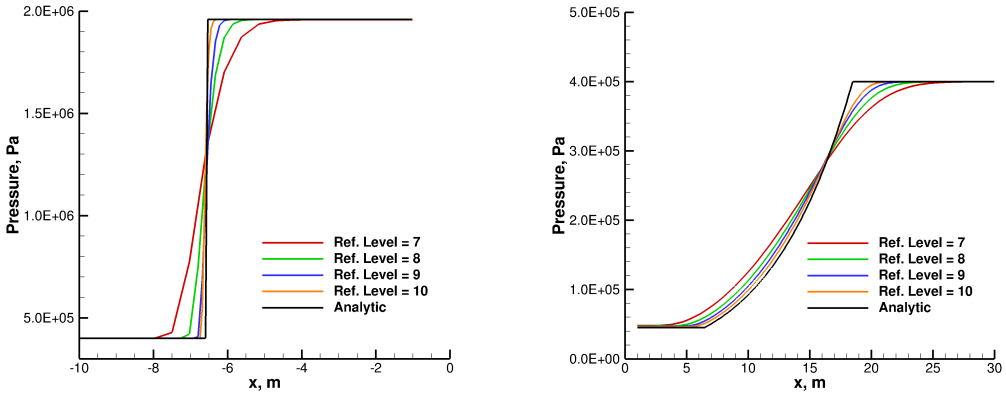
### 3.2.2 Discussion on Accuracy

Further investigation of the numerical accuracy of the piston simulation is discussed next. The previous section presented results for a finely resolved mesh. In this section, simulations are presented where the mesh resolution is varied, the order of accuracy for the inviscid flux scheme is explored, and the effects of piston velocity are studied. All geometry, grid dimensions, and numerical parameters are identical to the grid converged simulations of the previous section, unless otherwise specified.

First, a grid resolution study is performed, by varying the refinement level from a value of 7 to 10. Note that the cell size, $\Delta x$, can be computed as a function of the grid size, $L$, and the mesh refinement level as

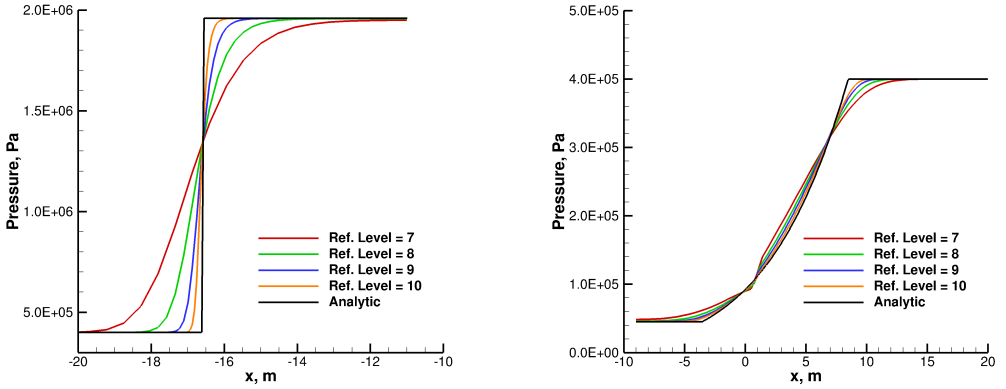
$$\Delta x = \frac{L}{2^n} \tag{28}$$

The grid resolution study looks at both stationary and translating piston simulations. The stationary piston is initialized such that the flowfield has a nonzero velocity of $1,000 m/s$, while for the translating piston simulations, the flowfield is initialized to a velocity of zero, and the piston travels at a speed of $1,000 m/s$. Comparisons are made between the numerical simulations and the analytic solution for each side of the piston. The pressure distributions for the stationary simulations are shown in Fig. 26. As expected, the numerical solution is most dissipative for coarser meshes. Similar plots are shown in Fig. 27 for the translating piston simulations. Note that the location of the shock and expansions waves are different, as they have traveled a different distance, compared to the stationary simulations. Likewise, the same trend of increasing accuracy with mesh refinement is seen. However, it is clear that there is significantly more dissipation across the shock in this case.




(a) Shock (b) Expansion

Figure 26: Pressure distribution through the shock wave and expansion, as a function of mesh refinement level (Ref. Level) for the stationary piston.



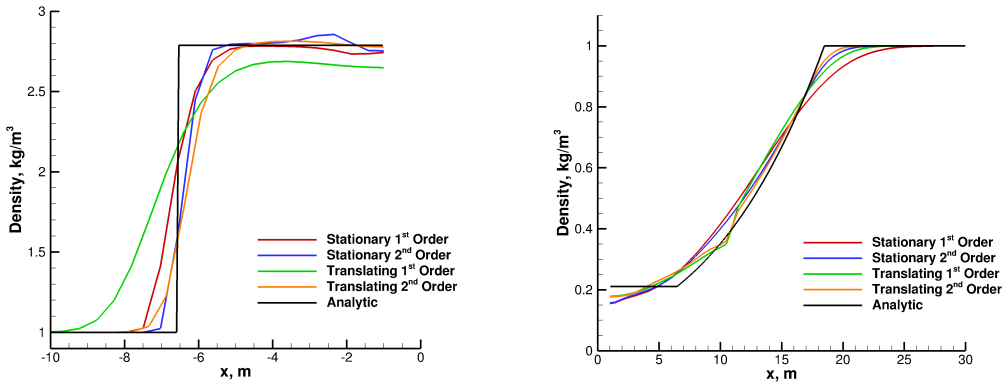

(a) Shock (b) Expansion

Figure 27: Pressure distribution through the shock wave and expansion as a function of mesh refinement level (Ref. Level) for the translating piston.

To explore this further, a single mesh refinement is considered (refinement level = 7), and the effects of order of accuracy are investigated. Density and pressure distributions are shown in Figs. 28 and 29, with both the shock and expansion cases considered. The density profile through the shock in Fig. 28a provides a clear picture for comparing the four approaches. From a purely qualitative standpoint, the accuracy of each approach can be gauged based on the amount of dissipation, evident by the smearing of the shock, as well as the accuracy of the post-shock density. The translating $1^{st}$ order accurate simulation is the

least accurate on both accounts. The other three simulations improve upon the accuracy of the gradient, as well as the post shock density value. The most accurate simulation from a qualitative standpoint is the $2^{nd}$ order accurate stationary piston.

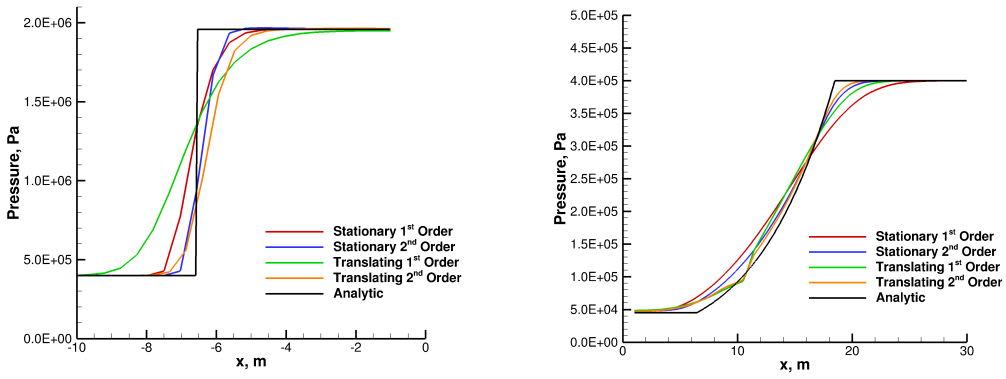
The expansion in Fig. 28b shows a similar inaccuracy near the piston, in terms of density magnitude. Additionally, there is also evidence of a numerical instability occurring where the slope in density abruptly changes. For the expansion, the translating $2^{nd}$ order accurate simulation follows the analytic solution the best, while the $1^{st}$ order accurate stationary simulation fairs the worst. These same trends can be seen in the pressure distributions in Fig. 29. However, do note that the pressure approaches the piston smoothly, and asymptotes towards the analytic solution. The accuracy in density and pressure approaching the piston correlates well with the expectations set forth during the previous discussion of the numerical overheating problem.




(a) Shock

(b) Expansion

Figure 28: Density distribution through shock wave and expansion for mesh refinement level 7. Comparisons are made between $1^{st}$ and $2^{nd}$ order spatial accuracy for the stationary and translating piston.




(a) Shock

(b) Expansion

Figure 29: Pressure distribution through shock wave and expansion for mesh refinement level 7. Comparisons are made between $1^{st}$ and $2^{nd}$ order spatial accuracy for the stationary and translating piston.

It is important to re-iterate that the most accurate solution on the compression side and the expansion side comes from using different strategies. However, this is natural when the numerical nature of the problem is considered. Despite the fact that the stationary and translating piston simulations present different numerical challenges, they are in fact representative of the same physical problem. This implies that the shock wave and expansion waves travel at the same velocity relative to the piston when comparing the two different frames of reference. Specifically, this implies the following: the shock and expansion waves generated by the stationary piston ideally travel at the theoretical wave speeds with respect to the piston, and with respect to the stationary background mesh, since both are at rest. When the piston is in motion, the shock and expansion waves still travel close to the theoretical wave speed with respect to the piston. However, the fact that the piston is in motion, implies that the waves are now traveling at different velocities with respect to the stationary mesh, when compared to those generated by the stationary piston.

This is important to consider from a numerical standpoint because artificial viscosity generated by Riemann solvers scales with wave speed. First, consider the compression side of the piston where the shock wave is generated. In the stationary piston simulation, the shock wave travels to the left at the shock speed, $u_s$, with respect to the piston and the

stationary mesh. Thus, the global velocity of the shock wave is $u_{s,g} = u_s$. In the translating piston simulation, the shock wave travels to the left at the shock speed relative to the piston, where $u_{s,g} = u_s + u_p$. The latter simulation results in a shock with a larger wave speed, leading to an increase in artificial viscosity, where the shock is highly dissipated and smeared over a larger distance.

When considering the expansion side of the piston, the opposite conclusion can be made. In the stationary piston simulation, the head of the expansion wave travels to the right at the local speed of sound $a_{initial}$. However, in the translating piston simulation, the motion of the piston towards the left reduces the global wave speed of the expansion propagating to the right. Thus, the lower global wave speed generated in the translating piston simulation results in less artificial viscosity when compared to the stationary piston simulation. The conclusion is that the artificial viscosity is minimized on the compression side when the stationary piston is considered, and is minimized on the expansion side when the translating piston is considered.

The numerical dissipation associated with the AUSMPW+ flux scheme can be quantified, and comparisons made between the stationary and translating piston simulations. The numerical dissipation on the compression side can be computed by considering the local region surrounding the shock wave. If the shock wave is aligned precisely with a cell interface, the pre-shock and post-shock conditions are known in the adjacent computational cells. A standard formulation for the numerical flux [58] can be written in terms of the numerical dissipation ($\vec{D}$) as

$$\vec{F}_{\frac{1}{2}} = \frac{1}{2}\left(u_L\vec{\phi}_L + u_R\vec{\phi}_R + \vec{P}_L + \vec{P}_R\right) + \vec{D} \tag{29}$$

The AUSMPW+ flux function [58] is computed as

$$\vec{F}_{\frac{1}{2},AUSMPW+} = \bar{M}_L^+ c_{\frac{1}{2}}\vec{\phi}_L + \bar{M}_R^- c_{\frac{1}{2}}\vec{\phi}_R + \left(P_L^+|_\alpha\vec{P}_L] + P_R^-|_\alpha\vec{P}_L\right) \tag{30}$$

Solving for $\vec{D}$, the numerical dissipation can be written as

$$\vec{D}_{\frac{1}{2}} = \bar{M}_L^+ c_{\frac{1}{2}} \vec{\phi}_L + \bar{M}_R^- c_{\frac{1}{2}} \vec{\phi}_R + \left( P_L^+ |_\alpha \vec{P}_L] + P_R^- |_\alpha \vec{P}_L \right) - \frac{1}{2} \left( u_L \vec{\phi}_L + u_R \vec{\phi}_R + \vec{P}_L + \vec{P}_R \right) \quad (31)$$

For the current physical problem under consideration, the thermodynamic state of the fluid on the left and right sides are the same between the stationary and translating piston simulations. The only difference is in the local flow velocity, as a result of the reference frame shift. The dissipation vectors associated with the stationary and translating piston simulations have been computed within the numerical flux scheme and are listed below. These vectors demonstrate the increase in magnitude of the numerical dissipation with piston velocity.

$$\vec{D}_{stationary} = \begin{bmatrix} 3.44 \times 10^2 kg/\left(m^2 \cdot s\right) \\ -5.69 \times 10^5 Pa \\ 1.05 \times 10^9 Pa \cdot m/s \end{bmatrix} \quad (32)$$

$$\vec{D}_{translating} = \begin{bmatrix} 1.31 \times 10^3 kg/\left(m^2 \cdot s\right) \\ -2.33 \times 10^6 Pa \\ 3.94 \times 10^9 Pa \cdot m/s \end{bmatrix} \quad (33)$$

Next, the numerical accuracy of these simulations is explored by performing mesh refinement studies, and examining the errors generated. First, it should be noted that the contribution of numerical error from the temporal integration is negligible for the time steps considered. The minimum time step requirements, based on numerical stability of the physical problem, are sufficiently small such that only a study of the spatial resolution is warranted. Several mesh refinements have been considered, and for each, four simulations are presented: 1) stationary piston with $1^{st}$ order accuracy, 2) stationary piston with $2^{nd}$ order accuracy, 3) translating piston with $1^{st}$ order accuracy, and 4) translating piston with $2^{nd}$ order accuracy.

Figures 30 and 31 depict the $L_1$ and $L_2$ errors in density for the shock and expansion sides of the piston. When performing a grid convergence study, the $L_1$ and $L_2$ errors must be normalized by the number of computational cells used in each simulation. A standard approach for calculating these values is done by using Eqns. 34 and 35.

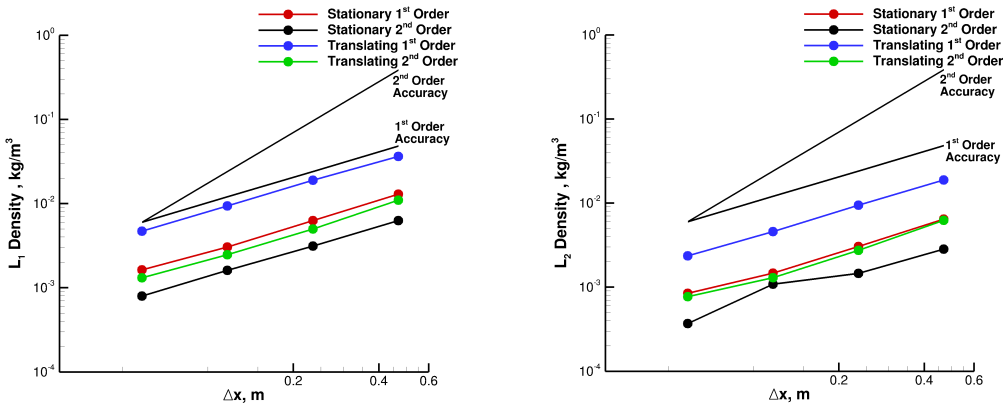
$$L_1 = \frac{1}{N} \sum_{i=1}^{N} |u_i - u_{exact}| \tag{34}$$

$$L_2 = \frac{1}{N} \sqrt{\sum_{i=1}^{N} (u_i - u_{exact})^2} \tag{35}$$

where $u_i$ refers to the quantity of interest within the computational cell of index $i$, and the summation is performed over $N$ computational cells. Additionally, in order to take variations over time into account, the average error in time is considered.

When comparing these errors between the stationary and translating piston simulations, there are additional concerns to explore. Consider the compression side, where the piston moves through the stationary mesh, and consumes computational cells as it sweeps across the domain. The total number of computational cells in the compression region is reduced. Additionally, the shock wave location sits further to the left than for the stationary piston simulation. This implies that the number of pre-shock computational cells has also reduced, compared to the stationary piston simulation. Thus, there is a much larger region of either zero or negligible contributions in error in the pre-shock stationary piston simulation. This effect artificially inflates the $L_1$ and $L_2$ error calculations of the translating piston.

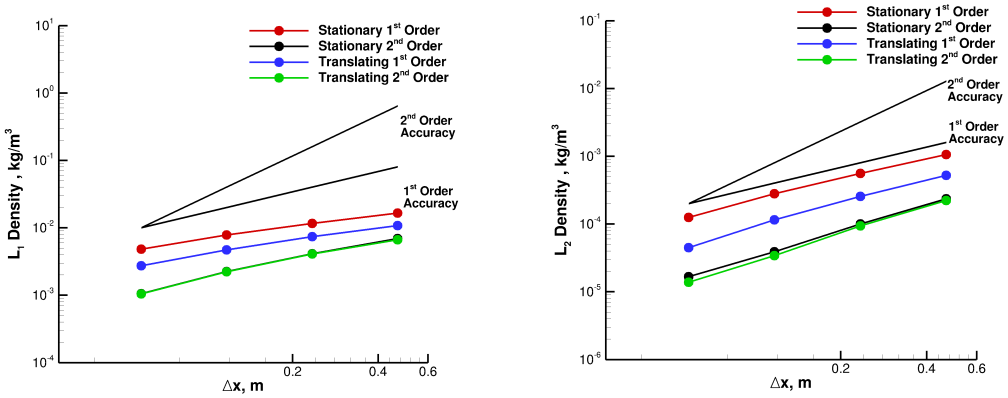
Thus, the approach taken is to let the value of $N$ be the total number of computational cells within the domain at the start of the simulation. Figure 30 shows approximately $1^{st}$ order accuracy on the compression side for the stationary piston simulations when considering both the $L_1$ and $L_2$ error. Theoretically, $2^{nd}$ order global accuracy cannot be achieved since the surface boundary conditions are limited to $1^{st}$ order accuracy. However, the $2^{nd}$ order accurate flux scheme does result in a decrease in error. The translating piston simulations also show $1^{st}$ order accuracy for both simulations. It has also been observed that the structured stencil-based approach used for these simulations results in $1^{st}$ order accuracy, despite the use of higher order schemes. A detailed discussion on this topic is presented in Appendix E.




(a) (b)

Figure 30: $L_1$ and $L_2$ error in density as a function of cell size for the compression side of the flowfield.

Order of accuracy plots for the expansion side of the piston are shown in Fig. 31. The stationary piston simulations result in larger error when compared to the translating piston simulation for each spatial scheme. Note the indistinguishable difference in $L_1$ error when comparing the stationary and translating $2^{nd}$ order accurate simulations. The $L_2$ error only marginally amplifies the differences in these simulations.



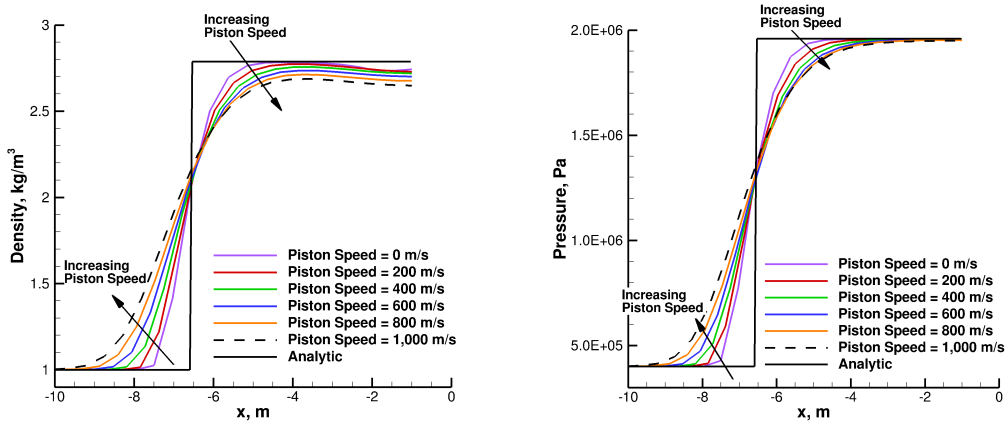

(a) (b)

Figure 31: $L_1$ and $L_2$ error in density as a function of cell size for the expansion side of the flowfield.

In order to further characterize the effects of piston velocity on the numerical results, a range of piston velocities have been considered. Inline with all simulations presented

thus far, comparisons are made such that the relative velocity between the piston and the fluid remains constant. That is, if the piston is stationary, the initial flowfield contains the entirety of the velocity contribution. On the other hand, if the piston is moving, then the flowfield velocity has been set to zero, while the entire velocity contribution is generated by the piston. In the following simulations, several piston velocities are considered ($u_p = 0m/s$, $u_p = 200m/s$, $u_p = 400m/s$, $u_p = 600m/s$, $u_p = 800m/s$, $u_p = 1,000m/s$). Thus, the corresponding flowfield velocities are initialized to a spectrum of values ($u_f = 1,000m/s$, $u_f = 800m/s$, $u_f = 600m/s$, $u_f = 400m/s$, $u_f = 200m/s$, $u_f = 0m/s$) respectively.

Density and pressure distributions through the shock wave are shown in Fig. 32, where the numerical solutions are compared against the exact analytic solution. As expected, the variation in piston velocity results in a smooth variation in density and pressure distributions, bounded by the limiting cases studied up to this point. These plots effectively demonstrate the reduction in error as the piston speed is increased.

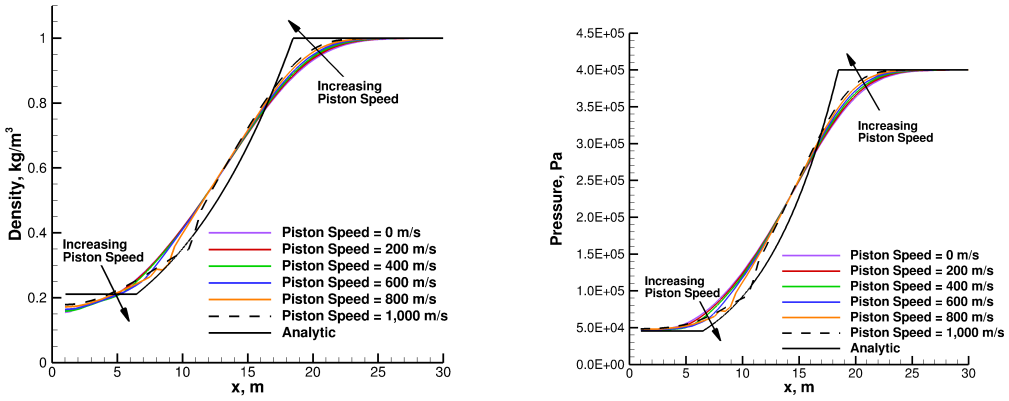



(a) (b)

Figure 32: Comparison between density and pressure distributions through the shock wave as a function of piston velocity, while relative velocity between piston and fluid remains constant.

The density and pressure distributions for the expansion side are shown in Fig. 33. These plots demonstrate how the solution accuracy is increased as the piston speed increases. A noteworthy observation is that the kink in the plots seems to disappear when the piston

velocity falls approximately below the speed of sound. This is further explored in Fig. 34, where this numerical artifact is clearly present when the piston is traveling at $700m/s$, but disappears below this velocity. It is possible that there is a connection between the piston velocity, relative to the local speed of sound causing this. However, further investigation is warranted, with particular attention towards the approaches used in setting the state vectors as cells emerge from inside the body.




(a) (b)

Figure 33: Comparison between density and pressure distributions through the expansion wave as a function of piston velocity, while relative velocity between piston and fluid remains constant.

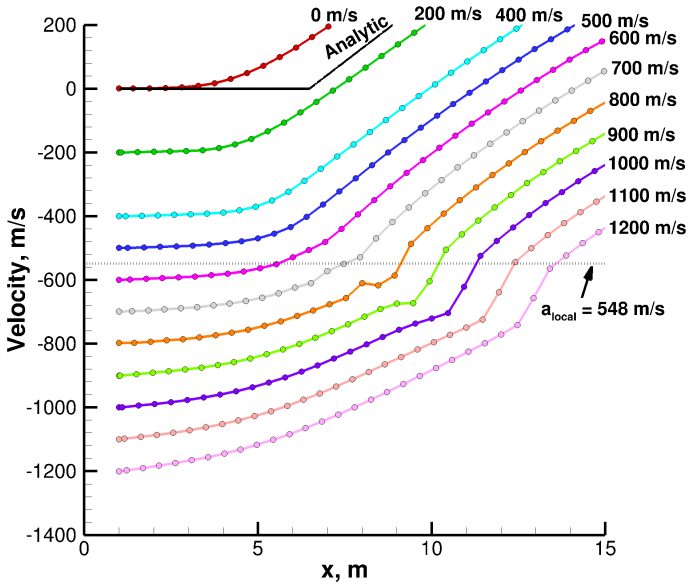



Figure 34: Velocity distribution in the region between piston and expansion waves. Axial position for each simulation is shifted such that piston location matches the stationary piston simulation. The velocity label along the right corresponds to piston speed, and the dashed line indicates the value of the local speed of sound at the foot of the expansion. Symbols represent mesh nodes.

Finally, these results are collected in Fig. 35, where the $L_1$ density error is plotted as a function of piston speed for two mesh refinement levels. The first characteristic of this plot demonstrates the reduction in error for the finer mesh. The second, highlights the two trends previously discussed for the shock and expansion cases. For the shock wave, the error increases with piston speed, while for the expansion wave, the error decreases with piston speed.

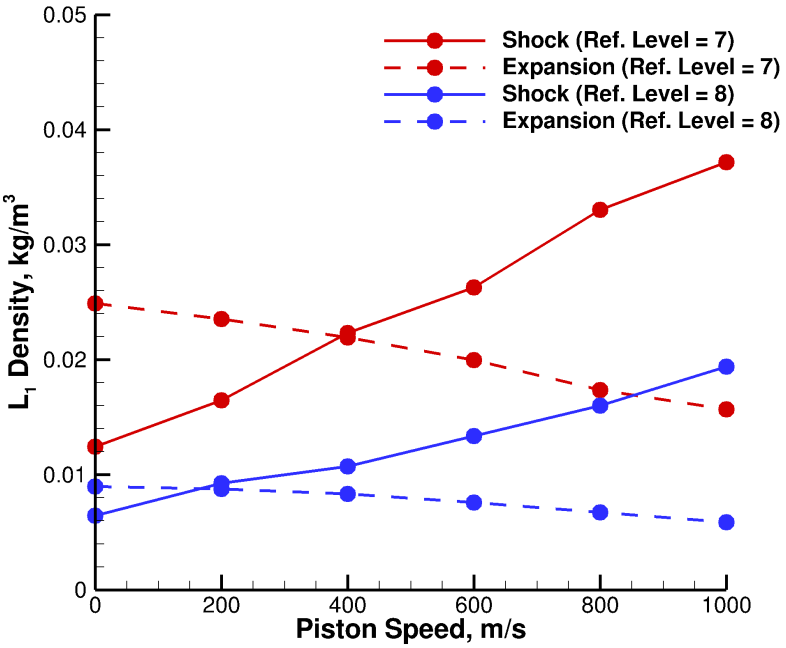



Figure 35: $L_1$ error in density as a function of piston velocity, while relative velocity between piston and fluid remains constant. Results are shown for both the shock and expansion regions for two mesh refinement levels.

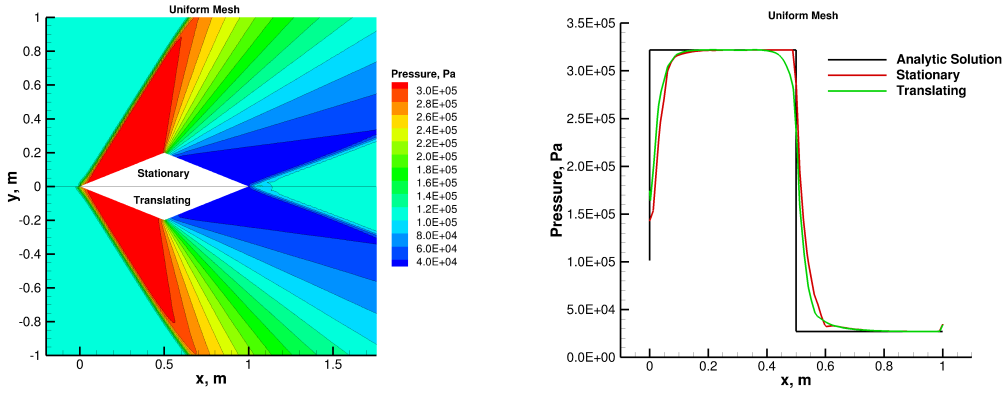
### *3.3 Supersonic Diamond Airfoil*

The first 2-D simulation is that of a supersonic diamond airfoil, and is chosen due to the simplicity of the flowfield it produces. An attached shock is generated at the leading and trailing edges, and an expansion fan at the mid-chord. This flow structure provides an analytic solution to the surface pressure distribution as the flow passes through these turn angles. The diamond airfoil has a 1 m chord length and a half angle of $21.8^o$. Two types of simulations are presented: 1) a stationary airfoil with Mach 2 flow and 2) an airfoil translating into quiescent flow at a velocity that corresponds to Mach 2 conditions. All simulations in this section were run using the AUSMPW+ scheme with $2^{nd}$ order accurate MUSCL extrapolation, and $1^{st}$ order time accuracy.

As with the previous discussion for the 1-D piston, the following analysis first considers a uniform Cartesian mesh. The resolution of the mesh has been chosen to be coarse enough to highlight differences between stationary and relative motion simulations. Since a uniform

mesh significantly increases the computational cost, the stationary simulation used a small computational domain size with supersonic outlet boundary conditions at all edges except for the freestream coming from the negative x-direction. In the relative motion simulation, the airfoil was required to travel approximately 35 m to achieve convergence. Running a uniform mesh of this size would be extremely costly. Thus, the domain size was set to the same size as the stationary simulation, and as the airfoil approached the edge of the computational domain, the domain was extended to ensure there was always sufficient computational space ahead of the airfoil. Likewise, the size of the domain in the wake was constantly reduced to eliminate cells that were not critical to the calculation. In all comparisons, the coordinates of the relative motion simulation are adjusted to coincide with the results of the stationary simulation.

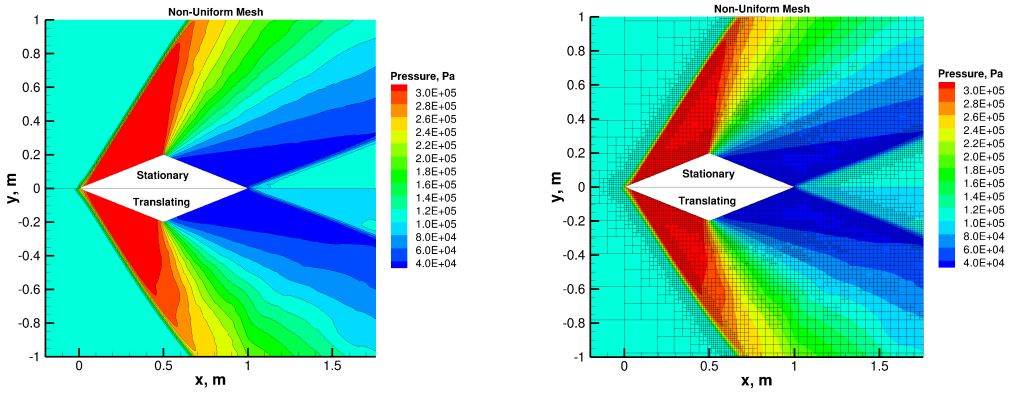
The results of the uniform mesh simulations are shown in Fig. 36. The flowfield shows a comparison of the pressure distribution between the stationary and translating simulations. The overall flow structures agree very well, and on close examination of the leading edge shocks, the effects of the moving body case are evident. The shock thickness is larger for the translating airfoil compared to the stationary case. A similar trend can be seen at the expansion fan as well, where the contour lines are more closely spaced for the stationary simulation. These results are more clearly seen in the plot of surface pressure distributions. In both cases, it is evident that the mesh resolution is not sufficient enough to capture the leading edge shock well. However, the purpose of this simulation was to show the differences in how well the shock and expansion fan was captured between the two frames of reference for a given resolution. For the translating simulation, the shock is thicker, resulting in a larger pressure by the time the flow arrives at the leading edge. In both cases, the pressure ultimately rises to the expected analytic value. The differences in capturing the expansion fan can clearly be seen in the surface pressure distribution.




(a) (b)

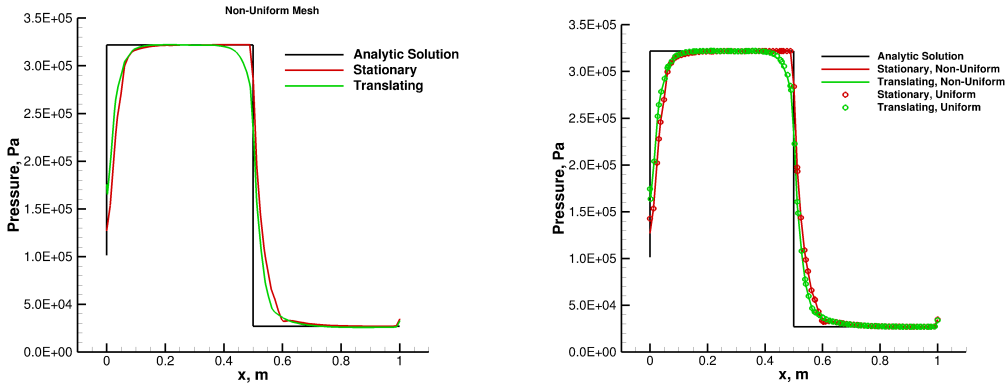
Figure 36: Comparison between a stationary and translating diamond airfoil on a uniform mesh. The pressure flowfield is shown in (a) and the surface pressure distribution is shown in (b).

Next, the same analysis was performed using a non-uniform mesh. For this particular problem, accuracy of the surface pressure distribution is primarily dependent on accurate shock capturing. This means that for all time in the simulation, the mesh must properly resolve the leading edge oblique shock. For the results presented, it was found that as long as the mesh maintained a thick enough band of fine cells around the shock, then this requirement was satisfied. Figure 37 shows comparisons between the two simulations. The oblique shocks compare well between the upper and lower half, while differences can be seen downstream of the shock. These differences are caused by the non-uniform mesh, though they do not impact the surface pressure distribution for this type of problem. The pressure distributions are shown in Fig. 38, where a similar discussion from the uniform mesh simulation can be made. It is evident that surface pressure distributions are nearly identical between the uniform and non-uniform meshes. This agreement is a strong function of how the entire flowfield affects the surface quantities. Since the surface quantities are driven by the leading edge shock and the centered expansion fan, it is only necessary to capture these phenomena accurately.




(a) (b)

Figure 37: Comparison between a stationary and translating diamond airfoil on a non-uniform mesh. The pressure flowfield is shown in (a), with the addition of mesh lines shown in (b).




(a) (b)

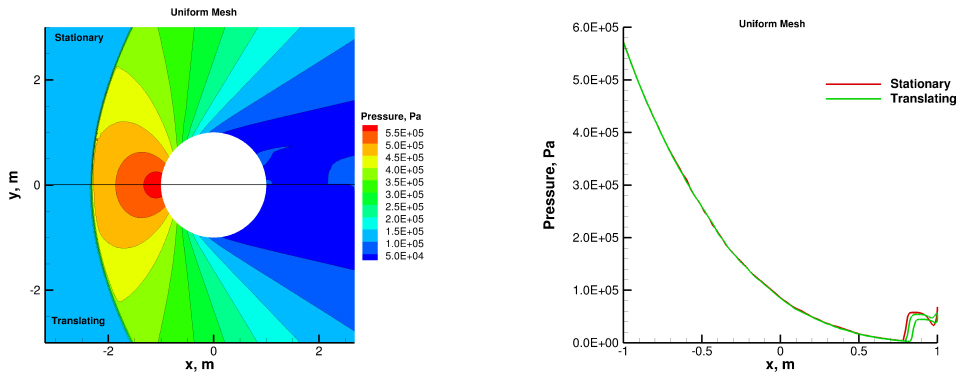
Figure 38: Comparison of surface pressure distribution between a stationary and translating diamond airfoil for a non-uniform mesh (a) and a comparison between all diamond airfoil results (b).

### *3.4 Supersonic Cylinder*

Supersonic flow around a cylinder is considered next. This configuration provides a very different flowfield in that there is a detached bow shock, producing a region of subsonic flow in front of the cylinder. Unlike the diamond airfoil, the flowfield properties are not constant behind the shock, which further tests the moving body implementation, as well

as the effects of non-uniform meshes. The relative Mach number between the cylinder and freestream is set to $M = 2$. Once again, the stationary simulation examines a geometry placed into Mach 2 flow, while in the translating simulation the cylinder moves at Mach 2 into quiescent flow. All simulations in this section were run using the AUSMPW+ scheme with $2^{nd}$ order accurate MUSCL extrapolation, and $1^{st}$ order time accuracy.

The first simulation presents a comparison on uniform meshes, as shown in Fig. 39. The flowfields are very similar, with differences primarily in the shock thickness, and disagreement in the wake. The shock thickness has increased due to the additional dissipation found in the translating simulation, as discussed in previous examples. Additionally, there are small oscillations present behind the shock in the stationary case, which are a well known phenomena often observed behind curved shocks that are not aligned with cell faces. The added dissipation in the translating simulation provides enough damping to eliminate most of the numerical post-shock oscillations. The differences in the wake region can be attributed to the issue of emerging cells. For the current mesh resolution, the states being set for emerged cells cannot exactly replicate the same physics being captured in the stationary simulation. The surface pressure distribution shows nearly perfect agreement along the cylinder, until a small region on the rear of the cylinder is approached.



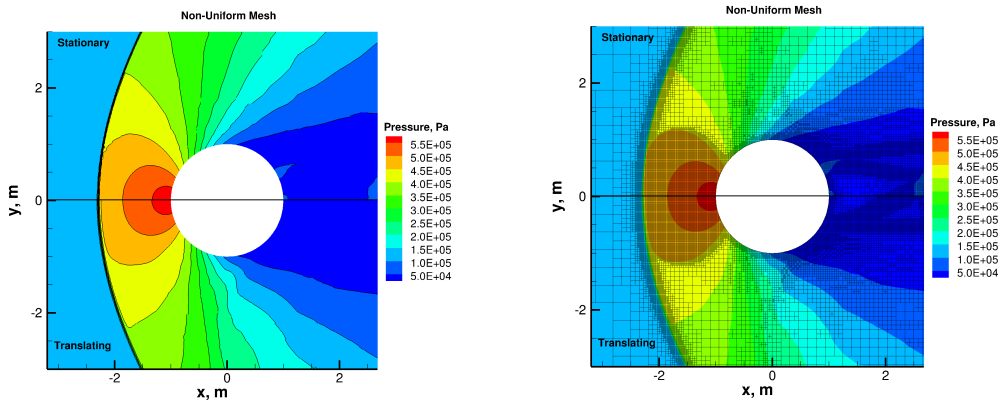

(a) (b)

Figure 39: Comparison between a stationary and translating supersonic cylinder on a uniform mesh. The pressure flowfield is shown in (a) and the surface pressure distribution is shown in (b).

Figure 40 illustrates the differences seen when considering a non-uniform mesh. As with any CFD simulation, the quality of the mesh plays a significant role in determining the accuracy of the solution. It has been found that extra care must be taken to ensure that the post-shock region of the mesh is well maintained, especially for the translating cylinder simulation. Performing solution-based grid adaption on the typical parameters such as velocity divergence, or gradients in the flow properties was found to be insufficient. Thus, an additional parameter was considered for this study. The absolute value for speed of sound was chosen, for which a cutoff value of $450m/s$ was chosen. This was chosen based on a priori knowledge of the flowfield, and cell refinement was set in regions of the flow where the speed of sound was greater than this value. Large values of speed of sound typically occur in the stagnation region for this type of simulation, and was thus a useful choice.

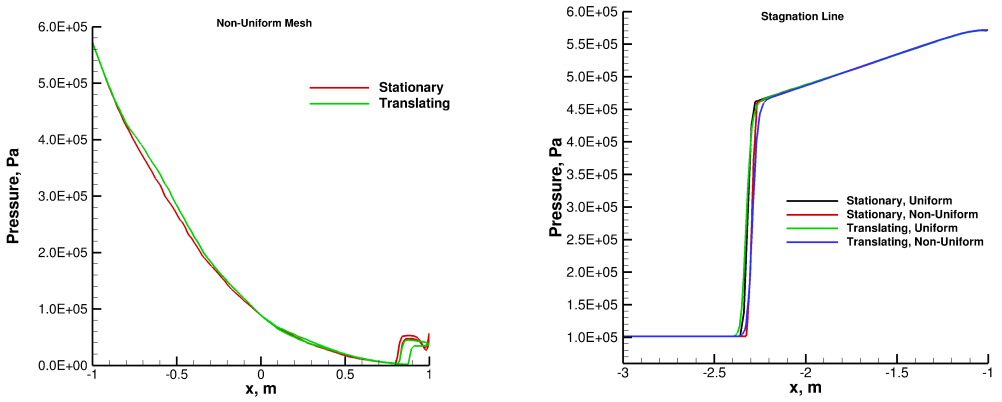
Since the mesh in the stagnation region is nearly identical between the stationary and translating simulations, the flowfield contours in this region appear to agree well. The contour lines show additional noise in the translating simulation in regions where the mesh is not sufficiently resolved. Figure 41 first shows a comparison between the surface pressure distribution for the non-uniform mesh simulations. The surface pressure between the two simulations is indistinguishable in the stagnation region, as desired, based on the similarity in meshes. However, as the surface streamlines leave the stagnation region, the pressure imparted on the cylinder begins to show minor deviations.

It is also interesting to compare the pressure distribution along the stagnation line. Such a plot provides a clearer picture as to how well the shock location is captured, and what the pressure looks like as the flow approaches the stagnation point of the cylinder. The stationary and translating uniform mesh simulations show very close agreement along the entire stagnation line. The only differences are due to the slightly dissipative shock. However, a larger contribution of error stems from the differences in the non-uniform vs. uniform mesh approach.




(a) (b)

Figure 40: Comparison between a stationary and translating supersonic cylinder on a non-uniform mesh. The pressure flowfield solution is shown in (a), along with the grid lines in (b).




(a) (b)

Figure 41: A comparison of surface pressure distribution between a stationary and translating supersonic cylinder on a non-uniform mesh is shown (a). Stagnation line pressure profiles are shown for all supersonic cylinder simulations in (b).

### *3.5 Transonic Oscillating NACA 0012 Airfoil*

The next simulation considers a NACA 0012 airfoil oscillating in pitch. Computational results are compared against AGARD experimental data [65]. The pitch oscillations are governed by the sinusoidal expression,

$$\alpha(t) = \alpha_m + \alpha_o sin(2\pi f t) \tag{36}$$

where $\alpha$ is the angle of attack, $\alpha_m$ is the mean angle of attack, $\alpha_o$ is the pitch oscillation amplitude, and $f$ is the pitch frequency. In a similar way that Mach number is the appropriate non-dimensional parameter for describing the convective behavior of the flowfield, reduced frequency provides the correct similarity parameter for matching unsteady aerodynamic phenomena. The reduced frequency, $K$, for simple harmonic motion is a function of pitch frequency, chord length, and freestream velocity.

$$K = \frac{\omega(c/2)}{V_\infty} = \frac{\pi f c}{V_\infty} \tag{37}$$

The case under consideration is identified in the AGARD report as CT Case 5. The flow and geometry parameters are listed below in Table 2. Two inviscid simulations were run: 1) relative body motion and 2) rigid grid motion through the use of the ALE formulation. Both simulations used the same grid dimensions spanning the range [-10m,10m] in both coordinate directions. The domain edge boundary conditions were set to the farfield condition to use Riemann invariants. Global, explicit time stepping was performed using the $3^{rd}$ order TVD Runge-Kutta scheme. Before initiating the pitching motion, the simulations were run to steady state at the mean angle of attack condition. The convective fluxes were computed using the AUSMPW+ scheme, along with the $3^{rd}$ WENO extrapolation scheme.

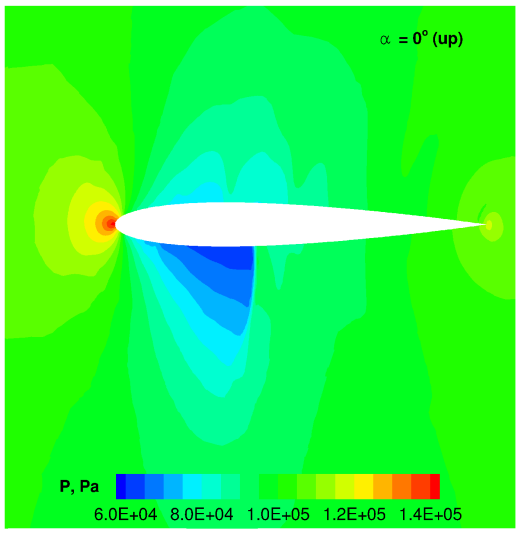
The relative motion simulation was computed with a constant time step of 5.0 x $10^{-7}$s. With an oscillating frequency of 62.5 hz, a single period takes 0.016 s, which corresponds to 32,000 iterations. The mesh resolution was set such that the airfoil had a minimum of 128 cells along the chord length, corresponding to a refinement level of 15 ($\Delta x = 20m/2^{15} \approx 0.61mm$). A uniform region of the finest cells was set around the airfoil up to a distance of 0.02 m. Solution adaption was turned on for the entire simulation at a frequency of 10 iterations. The velocity divergence was used as a means for performing solution adaption, with the coarsening and refining parameters set to 0.5 and 0.7 of the mean divergence respectively. The initial mesh had 15,740 computational cells, compared to the 21,660 at

Table 2: Geometry and flow conditions of AGARD CT Case 5.

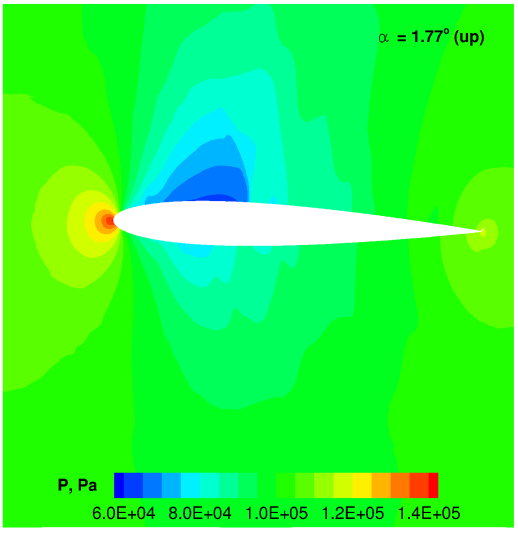
| Parameter | Value |
|---|---|
| $M_\infty$ | 0.755 |
| $V_\infty$ | 261.319 m/s |
| $\rho_\infty$ | 1.1841 kg/m$^3$ |
| $p_\infty$ | 101325 Pa |
| $T_\infty$ | 298.15 K |
| c | 0.1016 m |
| $\alpha_m$ | $0.016^o$ |
| $\alpha_o$ | $2.51^o$ |
| $f$ | 62.5 Hz |
| $K$ | 0.0814 |

completion. There were 370 computational surface cells that defined the airfoil geometry. The ALE simulation had all the same characteristics, with negligible changes in cell count.

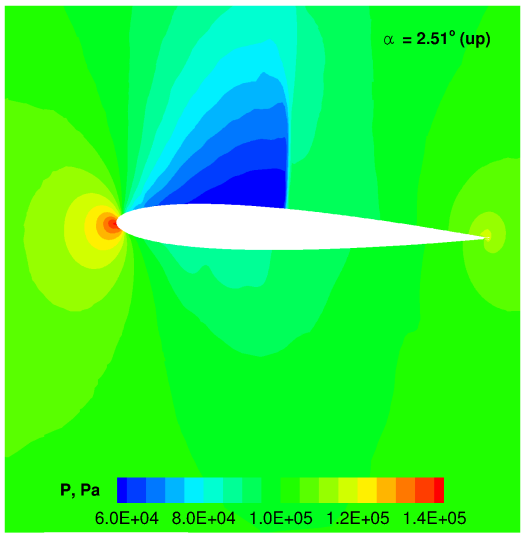
Snapshots of the flowfield are shown in Fig. 42 for four instances in time. The series starts with the airfoil pitching in a nose up direction at an angle of $0^o$. An intermediate angle of $1.77^o$ is shown, before the maximum pitch amplitude of $2.51^o$ is shown. The fourth image shows the airfoil pitching down as it passes through an angle of $1.77^o$. A plot of normal force coefficient is shown in Fig. 43. The ALE simulation represents the baseline solution given the current mesh resolution and numerical scheme properties. There is fairly good agreement between the ALE results and the experimental data. There is a trend of underprediction that has also be seen by several other researchers [83, 79, 71, 39, 52, 7], that is likely caused by the shock-boundary layer interaction that exists in the experiment, but not in the numerical simulations. Now, considering the relative motion simulation, there is good agreement compared to the ALE approach, with the largest deviations occurring at the maximum pitch up and pitch down conditions. The normal force coefficient curve for the relative motion case also produces slightly more noise as a result of the computational surface mesh constantly changing as the airfoil sweeps through the stationary Cartesian mesh.



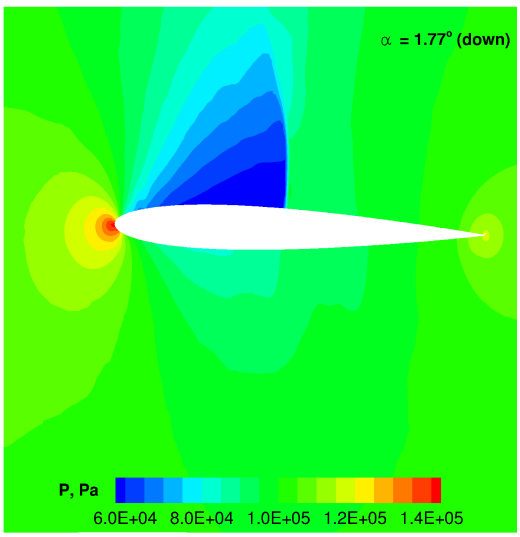

(a) $\alpha = 0^o$, Nose pitching up.




(b) $\alpha = 1.77^o$, Nose pitching up.




(c) $\alpha = 2.51^o$, Maximum pitch.




(d) $\alpha = 1.77^o$, Nose pitching down.

Figure 42: Snapshots of flowfield pressure distributions for the relative motion, oscillating NACA 0012 airfoil.

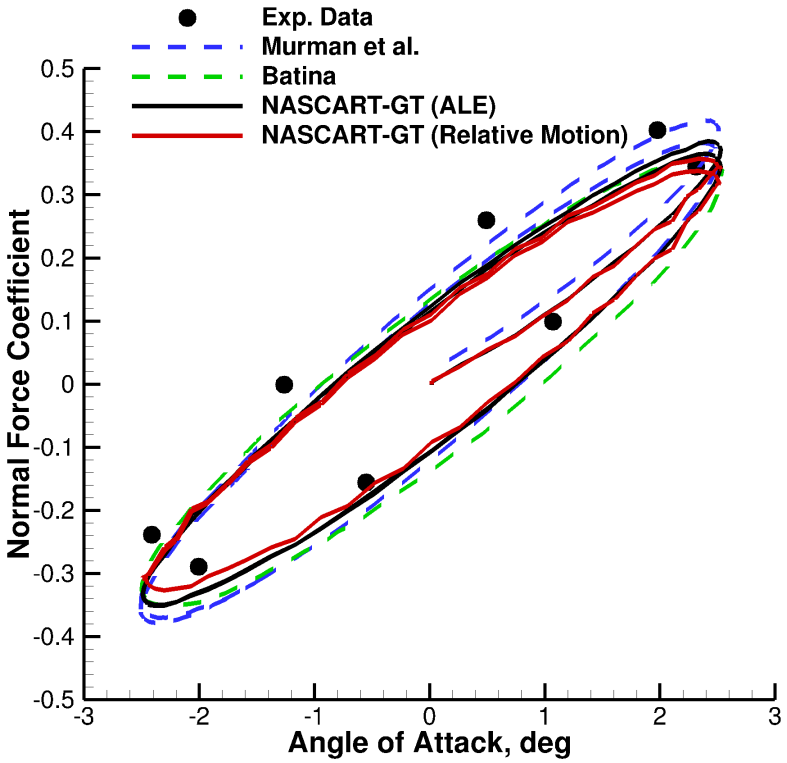



Figure 43: Normal force coefficient as a function of angle of attack for an oscillating NACA 0012 airfoil. Results are compared against experimental data from Ref. [65].

# CHAPTER IV

# FREE FLIGHT SIMULATIONS

Aerodynamic coupling with rigid body motion presents a class of problems that further advances the modeling towards time accurate FSI. The rigid body dynamics of interest in the present chapter are based on experimental tests performed in a supersonic ballistic range. As such, these types of analyses are typically referred to as free flight simulations. This class of problem is essentially a subset of the larger FSI problem, given that there is strong coupling between the fluid flow and structural behavior of the vehicle. In this particular case, however, the structure is not allowed to elastically deform. The only allowable degrees of freedom are the six rigid body motions.

In this chapter, coupling between the fluid dynamics and the 6-DoF model (Appendix B) is evaluated by both stationary and moving body problems, using a simplified 2-D MSL capsule, where comparisons are made against the results from US3D [87, 31]. The present work is based on a set of computational results [105] that model a series of ballistic range tests [98], studying the free flight motion of the MSL capsule. For the sake of computational efficiency, a 2-D simulation is investigated rather than a full 3-D analysis. Doing so reduces the computational time, and provides an efficient means for validation. Given that the problem is 2-D, the free flight dynamics will not replicate those of the 3-D ballistic range test. Thus, collaboration with the primary author of Ref. [105] provided a means for code-to-code comparisons. This section is broken down into three components. First, the physical problem is described, along with an overview of the numerical setup. Next, a series of static simulations are presented, which serve to validate the static aerodynamic loads. These analyses provide a baseline, with respect to spatial resolution, that can be used for the free flight simulations contained in the third section.

### *4.1 Physical Problem and Computational Mesh*

The 2-D MSL capsule has a diameter of $D = 4.5015m$, and is subjected to Mach 2.54 flow. The freestream density is $\rho_\infty = 2.88 \times 10^{-4} kg/m^3$, pressure $p_\infty = 20.31Pa$, temperature $T_\infty = 245.45K$, and velocity $V_\infty = 800.0m/s$.

The mesh quality in each of the simulations is a critical component in producing accurate simulations. The simulations presented in this chapter are compared against the numerical results using US3D, so it is important that the grids exhibit similar resolution and overall mesh quality. The dimensions of the NASCART-GT mesh are larger than that of the US3D mesh, which was done for the sake of ensuring that the boundaries were sufficiently far from the geometry, and that the bow shock could be resolved to a greater distance. A comparison between the two meshes can be seen in Fig. 44, where the NASCART-GT mesh has been adapted based on solution gradients, tracking the primary flow features, which have manifested as shock waves.

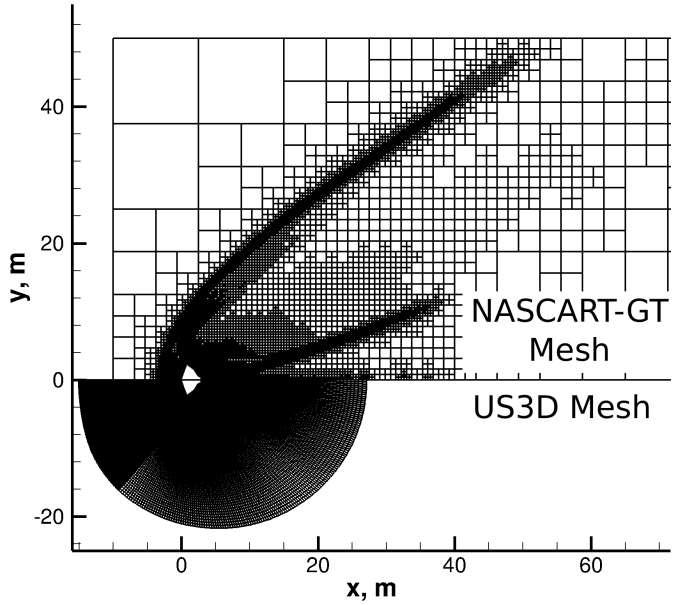
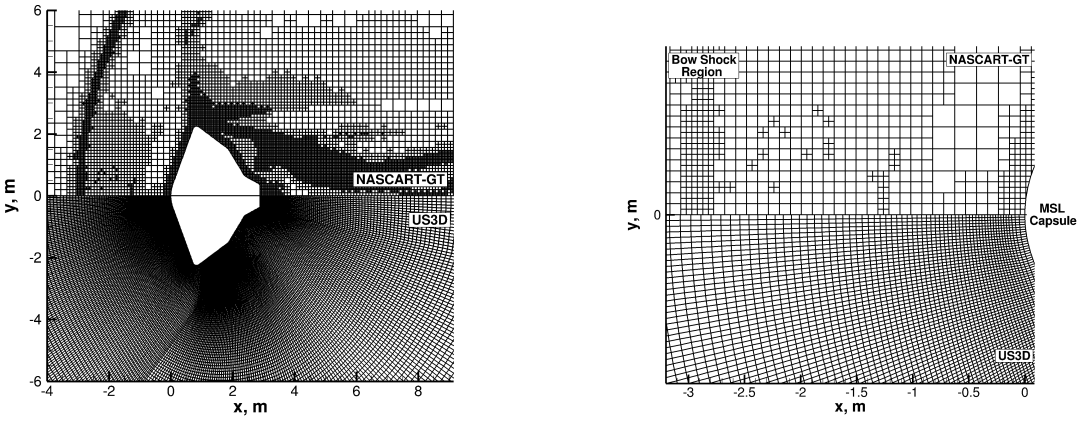



Figure 44: Comparison of meshes between NASCART-GT and US3D.

The mesh spacing for the present analyses has been chosen such that it is similar to the mesh used for US3D. Since the Cartesian mesh requires computational cells with an aspect ratio of unity, the two meshes cannot be perfectly matched in terms of cell size. Similarly

sized computational cells are most important to find near the surface of the geometry, as well as in the region of the bow shock. The smallest cell sizes in the NASCART-GT mesh have dimensions, $\Delta x = \Delta y = 0.049m$, which can be found at the surface of the geometry, as well as the shock region. The mesh used in the US3D simulations contains cells near the surface of size $\Delta x_n \approx 0.05m$ and $\Delta x_t \approx 0.022m$, while in the region of the bow shock, cell sizes are $\Delta x_n \approx 0.0881m$ and $\Delta x_t \approx 0.048m$, where $\Delta x_n$ is the dimension normal to the surface/shock and $\Delta x_t$ is tangential to the surface/shock.

Figure 45 illustrates two additional comparisons between the US3D and NASCART-GT meshes. In Fig. 45a, the major differences in meshing become apparent. Despite the fact that the computational cells near the surface and bow shock are comparable (as seen in Fig. 45b), there are stark differences within the shock layer, as well as the wake. Provided that there is no significant losses in solution accuracy, these regions of coarse cells in the NASCART-GT grid are highly desirable. They lead to a reduction in cell count, and are the direct result of efficient use of the solution adaption capability.




(a) (b)

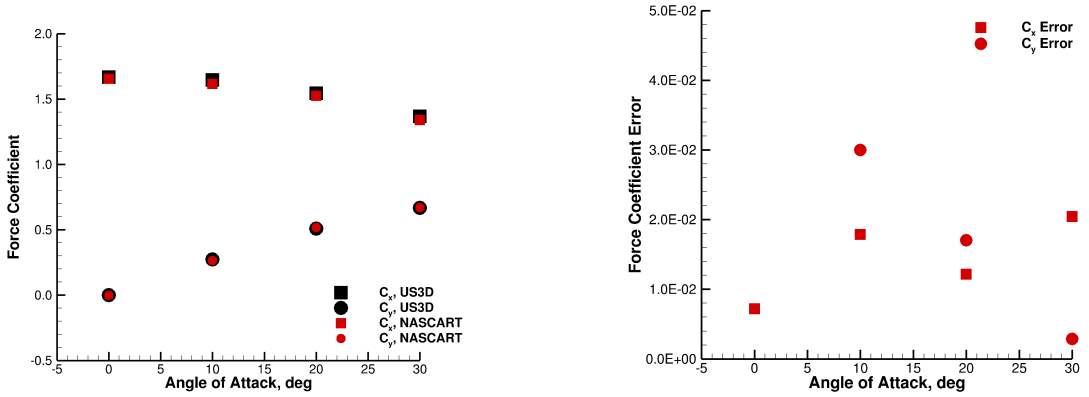
Figure 45: Comparison of meshes between NASCART-GT and US3D. The mesh in (a) highlights the differences within the shock layer, and the mesh in (b) compares the cell sizes near the surface of the capsule and in the region of the bow shock.

## *4.2 Static Analysis Verifications*

As a means of verifying NASCART-GT against US3D, a series of static simulations were run, spanning a relevant range of pitch angles. Four different angles of attack were run, including $0^o$, $10^o$, $20^o$, and $30^o$. The primary quantity of interest is the aerodynamic load since this is what governs the free flight motion in a dynamic simulation. The $3^{rd}$ order accurate TVD Runge-Kutta time scheme was used with a constant CFL of 1.0. The AUSMPW+ scheme was used with $1^{st}$ order accurate face extrapolation for the inviscid fluxes. Solution adaption was performed based on velocity divergence every 100 iterations until convergence was achieved. The Riemann Invariant boundary conditions were used on all edges of the grid boundary, and the slip wall surface boundary condition was used.

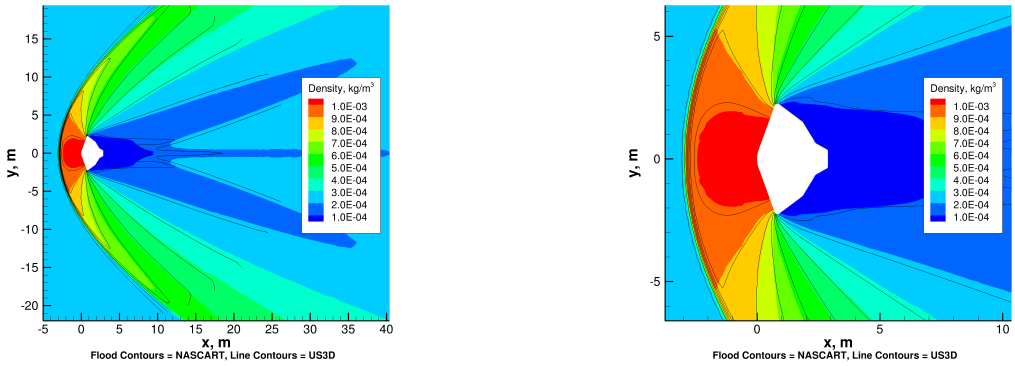
Fig. 46 shows a comparison of the force coefficient in the x and y directions as a function of angle of attack. It is clear that the aerodynamic characteristics computed using both flow solvers are in close agreement. The absolute value in error for the force coefficient quantifies this agreement, with a maximum relative error of less than 3%.

Several data sets are presented in Figs. 47 - 54 that provide more detailed information regarding each of the static simulations. All angles of attack are presented along with their flowfield density distributions and surface pressure coefficients. For each flowfield image, the NASCART-GT contours are shown by colored contouring, while the US3D contours are shown by lines. This provides an effective means for comparing the two flowfields. In general, the contours are much smoother for the US3D results, primarily due to the increased mesh quality. The figures showing the entire flowfield highlight the qualitative agreement in flowfield solutions, when considering a feature such as the bow shock. As the angle of attack increases, the differences in flowfield solutions become more evident. The plots of surface pressure coefficient show in greater detail where the solutions differ with respect to the surface location. Despite these differences, the integrated loads compare well, as quantified in Fig. 46.




(a) (b)

Figure 46: Aerodynamic load comparisons between NASCART-GT and US3D for the 2-D MSL verification simulations. The force coefficient is shown in (a) and the absolute force coefficient error is shown in (b).




(a) (b)

Figure 47: Flowfield comparison between NASCART-GT and US3D for $\alpha = 0^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours.

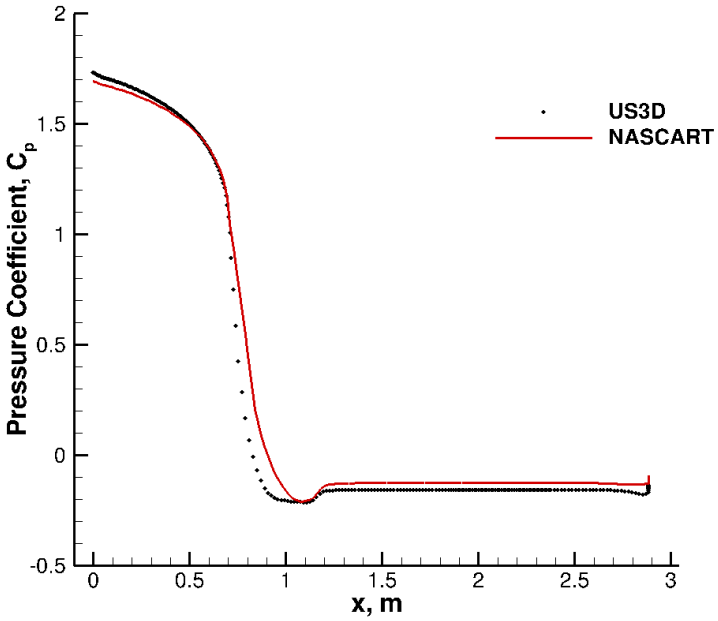
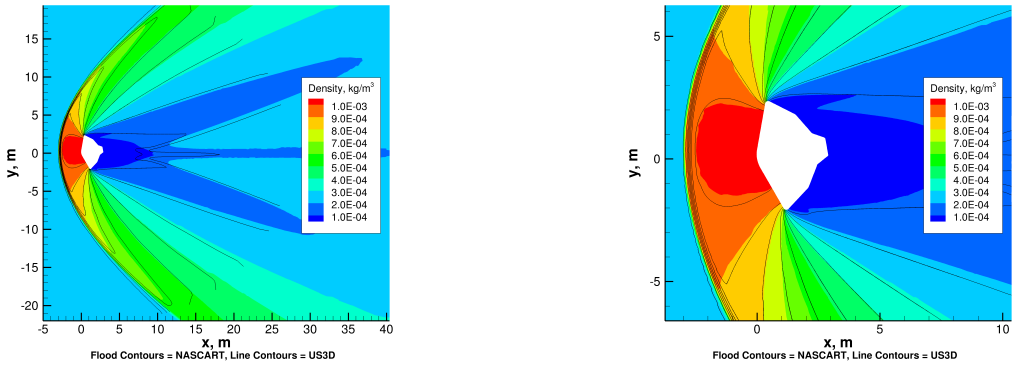



Figure 48: Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 0^o$ simulation.




(a) (b)

Figure 49: Flowfield comparison between NASCART-GT and US3D for $\alpha = 10^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours.

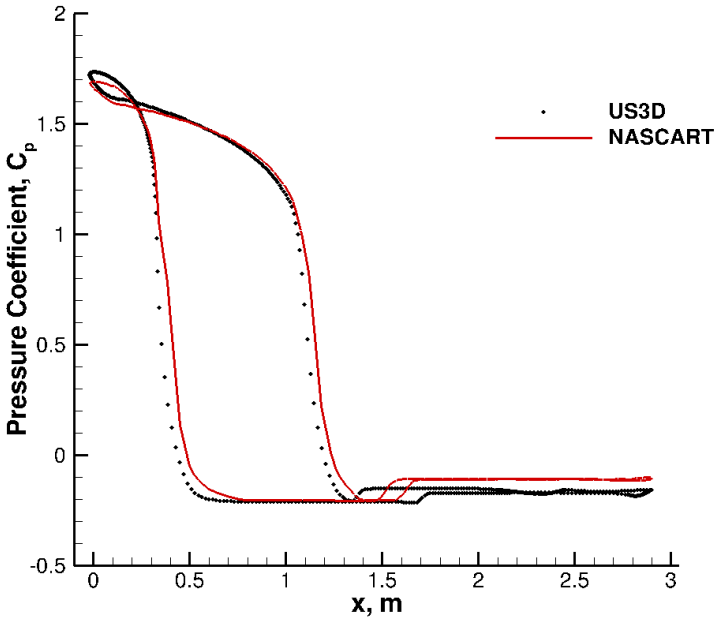
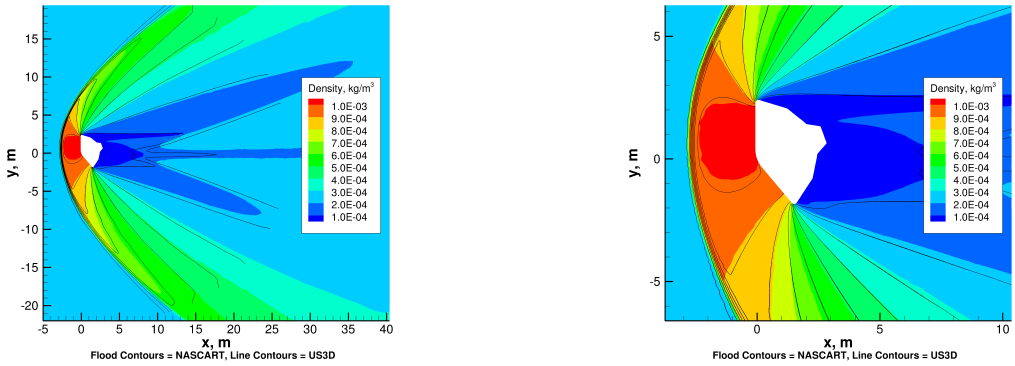



Figure 50: Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 10^o$ simulation.




(a) (b)

Figure 51: Flowfield comparison between NASCART-GT and US3D for $\alpha = 20^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours.

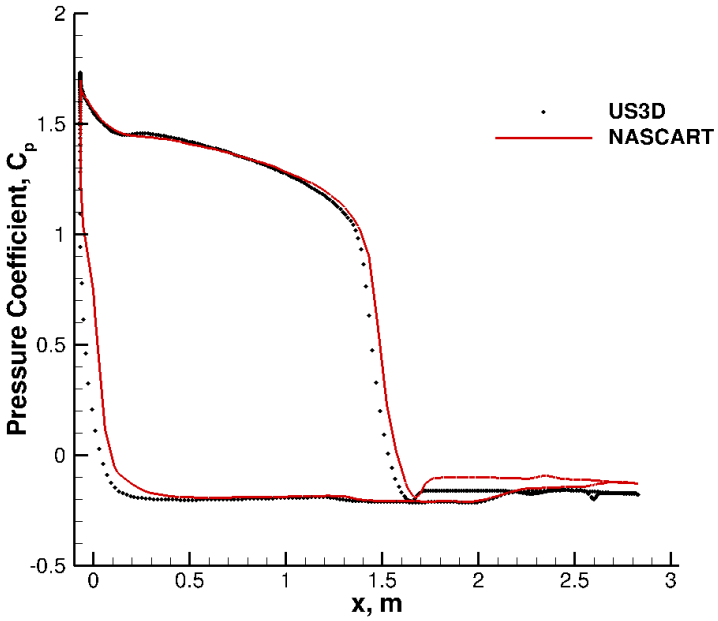
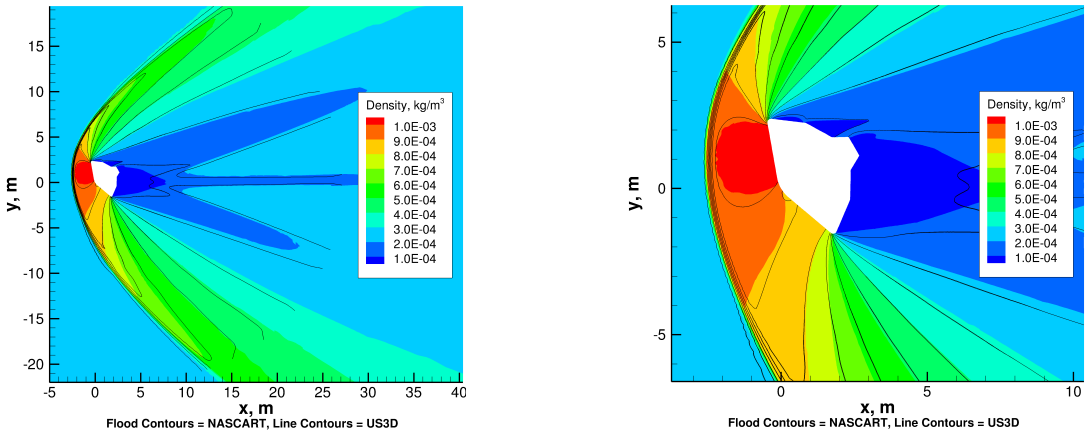



Figure 52: Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 20^o$ simulation.




(a) (b)

Figure 53: Flowfield comparison between NASCART-GT and US3D for $\alpha = 30^o$ simulation. The NASCART-GT solution is shown in colored density contours, while the US3D solution is shown in line contours.

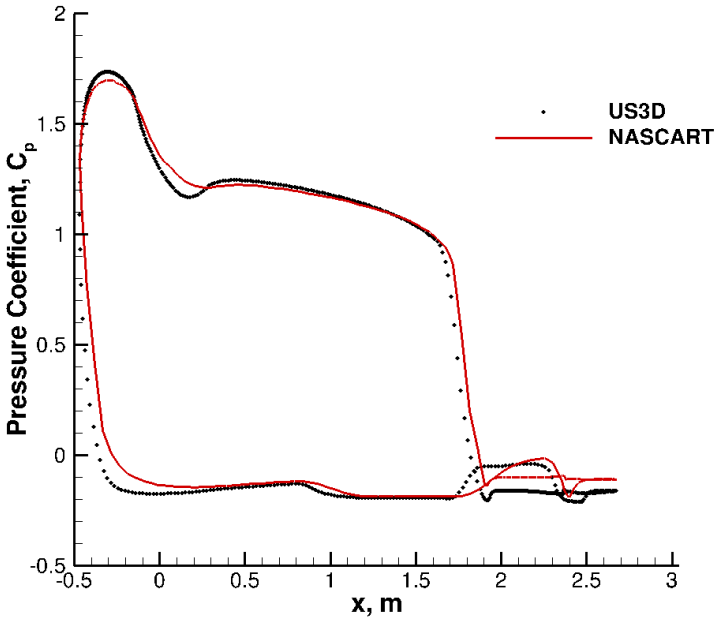



Figure 54: Comparison of surface pressure coefficient between NASCART-GT and US3D for the $\alpha = 30^o$ simulation.

### *4.3 Free Flight Analysis*

The free flight analysis currently considers a single release angle of $10^o$. The 2-D MSL capsule is constrained to planar motion, and fixed to its center of mass. A 1 DoF simulation is studied, where the single degree of freedom is rotation about the pitch axis. The center of mass is located at 0.299 capsule diameters along the axial direction from the nose, and the moment of inertia about the rotation axis is 2343.6 $kg \cdot m^3$, chosen to provide suitable dynamics for a verification analysis. Upon release, the aerodynamic loads applied to the vehicle were such that rotation towards $0^o$ is triggered. In a purely non-dissipative simulation, the capsule would be free to oscillate indefinitely with no loss in rotation amplitude. Despite the lack of viscosity in the Euler simulation, numerically generated artificial viscosity provides a non-physical means for pitch damping.

A comparison of pitch angle is shown in Fig. 55, where there are three NASCART-GT simulations compared against the US3D simulation. A constant time step of $\Delta t = 5.0 \times 10^{-5} s$ was used. The first NASCART-GT simulation is identified as using a similar mesh to US3D. This specifically means that the computational cells near the surface and bow shock are of the same approximate size. The cell dimensions are the same as described for the static simulations. However, a uniform region of cells has been included around the capsule with the goal of creating a relatively uniform region of cells out to a distance of $4m$. Additionally, no solution-based grid adaption was used. The purpose of this simulation was to ensure that the meshes between the NASCART-GT simulation and the US3D simulation were as close as possible, given their inherent differences. The second NASCART-GT solution reduced the region of uniform cells to only $1m$ away from the capsule. Solution-based grid adaption was used to refine the mesh in regions of large gradients, particularly in the vicinity of the bow shock. This approach ensured that the bow shock was resolved to a greater distance away from the body, compared to the first approach. Finally, the third simulation utilized a uniform region of cells that extended to a distance of $10m$ away from the capsule. Though this is not strictly a uniform mesh, all non-uniform cells are outside the spatial region that influences the surface of the capsule. For this reason, the simulation is referred to as being a uniform mesh.

The first observation is that the NASCART-GT simulations exhibit very little pitch damping, compared to the US3D simulation. It is noted that the CFL number was observed to play a role in pitch damping, where larger CFL numbers led to greater pitch damping. The US3D simulation was run using an implicit time scheme with CFL numbers up to 50. The explicit time stepping used in the NASCART-GT simulations put a constraint on the CFL number of approximately 0.5, which is theorized to be the reason for the reduced pitch damping.

The oscillation frequency of the first NASCART-GT simulation shows the best qualitative agreement with US3D. The frequency then increases for the remaining two NASCART-GT simulations, which is quantitatively shown in Table 3. This simply implies that the uniform mesh simulation yields a solution that demonstrates higher convergence, and is potentially more accurate.

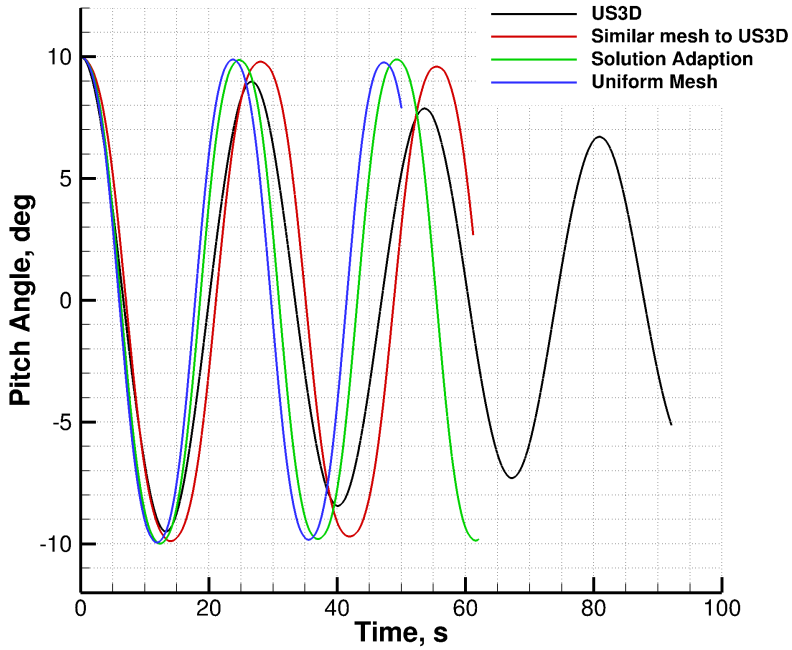



Figure 55: Comparison of meshing strategies of NASCART-GT against US3D simulation.

A grid convergence study is performed, where three mesh refinement levels are examined. The mesh refinement level refers to the mesh resolution near the surface of the capsule and region of mesh adaption along the bow shock. The surface refinement level of 16 corresponds

Table 3: Comparison of pitch frequencies.

| Simulation | Freq. (Hz) |
|---|---|
| US3D | $3.75 \times 10^{-2}$ |
| Similar mesh to US3D | $3.56 \times 10^{-2}$ |
| Solution Adaption | $4.04 \times 10^{-2}$ |
| Uniform Mesh | $4.22 \times 10^{-2}$ |

to the "Solution Adaption" simulation from Fig. 55. Two additional simulations were run, with the smallest cells half and one quarter the size respectively. The pitch angle is shown in Fig. 56. A geometry grid resolution of 32 is able to nearly capture the same frequency as the simulation with a resolution of 64, as can also be seen in Table 4. Also, note that there is greater pitch damping observed in the simulation with the finest mesh resolution. The time steps were kept constant, while the mesh resolution was increased. This results in a larger CFL number. Further investigation into smaller time steps may be of interest in future work, though was not studied in the present work due to the significant amount of computational time required for such a simulation.

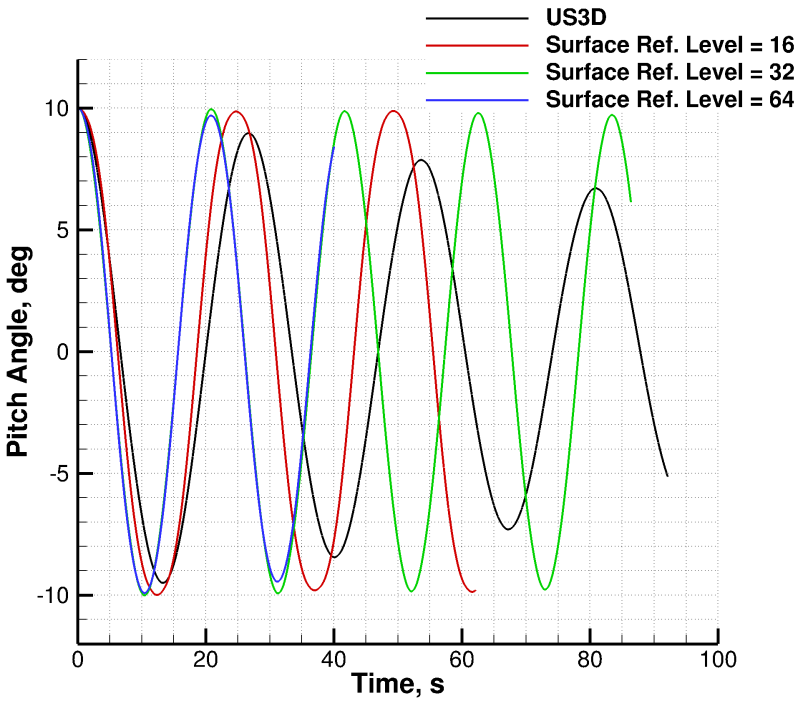



Figure 56: Grid convergence study.

Table 4: Comparison of pitch frequencies for grid convergence study.

| Grid Resolution | Freq. (Hz) |
|---|---|
| 16 | $4.04 \times 10^{-2}$ |
| 32 | $4.79 \times 10^{-2}$ |
| 64 | $4.80 \times 10^{-2}$ |

Using the converged simulation with a grid resolution of 32, the x and y force coefficient histories have been plotted in Fig. 57. The forces in the y-direction are the driving mechanism behind the pitching motion, and it can be seen that these loads are in phase with the response of the vehicle.

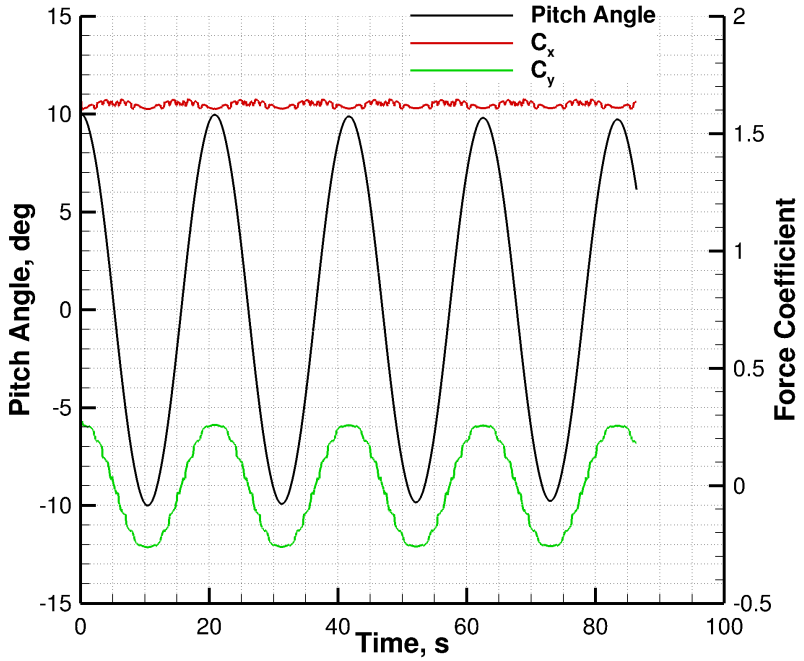



Figure 57: Grid converged simulation, showing the pitch angle history, as well as the time history of the force coefficients in the x and y directions.

# CHAPTER V

# FLUID-STRUCTURE INTERACTIONS

Thus far, all analyses have considered the coupling of rigid body motion with fluid dynamics. The final series of analyses investigates the interaction with elastic bodies. These fluid-structure interaction simulations are presented by investigating 1-D, 2-D, and 3-D problems. The focus of all simulations is on the time accurate nature of the problem, with steady-state coupling presented for the 3-D tension cone as a means of model validation. One dimensional elastic piston simulations present a simplified model for studying the numerical properties of the approach, and validating the methodology. Two dimensional simulations increase the complexity of the flowfield by considering the vortex induced vibrations of an elastic cylinder and the deformations of a beam subjected to supersonic flow. Finally, three dimensional simulations of a tension cone add significant complexity with respect to meshing, boundary conditions, and computational cost, but lay the groundwork for studying practical configurations of interest to the aerospace community.

## *5.1 Dynamic Analysis of Elastic Piston*

The model problem consists of an elastic piston, where one end of the domain is closed with a rigid wall, while at the opposite end, an elastic piston moves as a function of the applied fluid dynamic load. A diagram of the configuration is shown in Fig. 58. The problem is modeled in one spatial dimension and is a function of time. The fluid chamber is filled with a compressible fluid obeying the perfect gas law, and the flow is assumed to be inviscid. The initial length (L) of the fluid chamber is measured from the stationary end at the left hand side to the equilibrium position of the piston. The elastic piston is modeled as a discrete mass-spring system. The piston is represented as a point and has mass $m$, and the spring is assumed massless with a stiffness constant, $k$. This problem provides a simplified test case to analyze the numerical stability and accuracy of the coupled scheme.

The discrete piston system is governed by the harmonic oscillator equation:

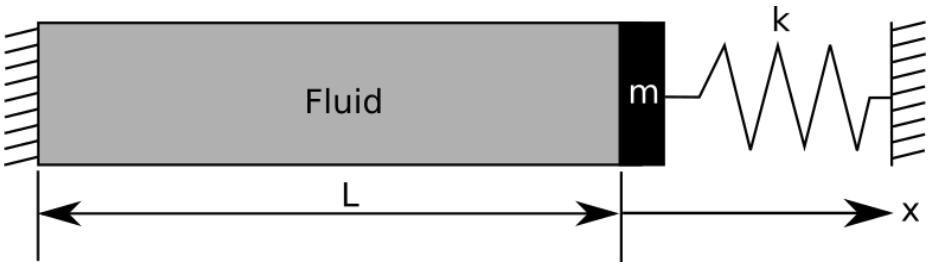



Figure 58: Model diagram of 1-D elastic piston configuration.

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = f(u,t) \tag{38}$$

where the damping coefficient, $c$, is assumed to be zero for the current analysis. $u$ is the displacement of the piston with respect to the equilibrium position, and $\dot{u}$ and $\ddot{u}$ are the velocity and acceleration of the piston respectively. The forcing function, $f(u,t)$, can either be prescribed, or is a result of the fluid dynamic loads.

The solution to this equation of motion is carried out using one of two options: 1) an algorithm implemented within NASCART-GT or 2) through the use of LS-DYNA. The following sections consider these two approaches in the analysis. Details regarding the validation of NASCART-GT's harmonic oscillator functionality can be found in Appendix C.

The mass of the piston is $0.8kg$ and the structural frequency is set to $\omega_s = 100rad/s$. The spring stiffness can be computed as $k = m\omega_s^2 = 8,000N/m$. The mass is given an initial velocity of $20m/s$. The initial flow conditions are set to $\rho = 1.3kg/m^3$, $a = 330m/s$, $P = 1atm$, $\gamma = 1.4$, and the initial size of the flow domain is $L = 1m$.

### 5.1.1 Elastic Piston with 0-D Fluid Model

The FSI problem is first investigated by considering a 0-D fluid model, where there are no spatial variations in flow properties. Such a model provides a simplified expression for evaluating the pressure applied to the surface of the piston, without the need for generating a CFD mesh, and solving the Euler equations. However, this comes at a loss in accurately capturing the true physics of the flow. Since spatial variations in flow properties are ignored, pressure waves created by the piston are not modeled. Nevertheless, the model provides

an effective baseline for validating the coupled problem. The fluid is assumed to be an adiabatic, perfect gas governed by the following relationship.

$$p\left(t\right)V\left(t\right)^{\gamma} = P_o V_o^{\gamma} \tag{39}$$

where $p\left(t\right)$ is the time varying pressure of the fluid, $V\left(t\right)$ is the time varying volume of the fluid chamber, and $\gamma$ is the ratio of specific heats. The volume of the chamber can be expressed as a function of the piston displacement:

$$V\left(t\right) = V_o + Au\left(t\right) \tag{40}$$

where $A$ is the cross-sectional area of the chamber and $V_o$ is the initial chamber volume. Assuming a chamber height and depth of unity, the cross-sectional area becomes one square unit. Additionally, the volume can be expressed strictly in terms of length. Thus, for the problem under consideration

$$V\left(t\right) = L\left(t\right) = L_o + u\left(t\right) \tag{41}$$

Substituting Eq. [41] into Eq. [39], and solving for $p\left(t\right)$ yields

$$p\left(t\right) = p_o\left(\frac{L_o}{L_o + u\left(t\right)}\right)^{\gamma} \tag{42}$$

and this expression is exactly equal to the applied force, $f(u)$, since $A = 1m^2$. Within the context of the following analyses, this expression is referred to as the nonlinear 0-D force expression. As an additional simplification for the following analyses, Eq. [42] can also be linearized for small displacements, resulting in the following linear 0-D forcing function.

$$p\left(t\right) = \left(\frac{-\gamma p_o}{L_o}\right)u\left(t\right) \tag{43}$$

Using the linearized pressure function in Eq. [38] with a damping coefficient of zero leads to the following governing equation.

$$m\ddot{u}\left(t\right) + ku\left(t\right) = \left(\frac{-\gamma p_o}{L_o}\right)u\left(t\right) \tag{44}$$

By linearizing the forcing function, the equation can be re-written such that the external fluid force can be absorbed into the spring constant term as

$$\ddot{u}(t) + \omega_n^2 u(t) = 0 \tag{45}$$

where $\omega_n$ is the natural frequency

$$\omega_n = \sqrt{\frac{k_{eff}}{m}} \tag{46}$$

and $k_{eff}$ is the effective spring constant for the equivalent unforced system,

$$k_{eff} = k + \frac{Ap_o\gamma}{L_o} \tag{47}$$

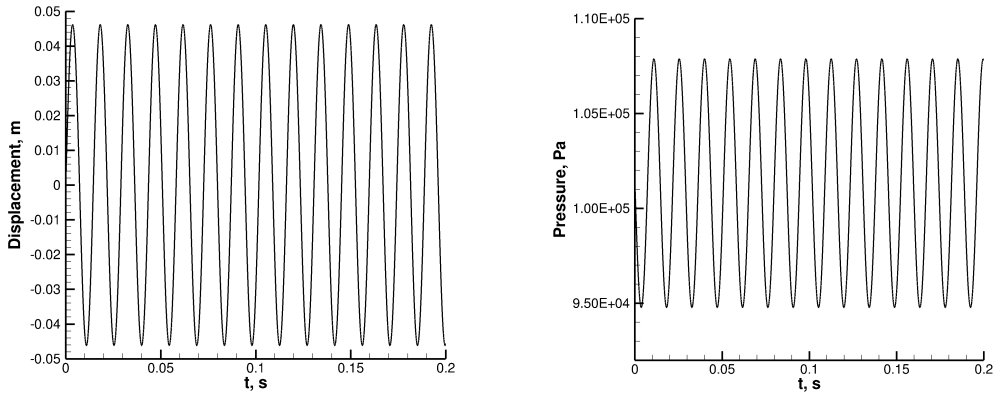
Thus, the analytic solution to this equation is

$$u(t) = u_o cos(\omega_n t) + \frac{\dot{u}_o}{\omega_n} sin(\omega_n t) \tag{48}$$

and

$$p(t) = p_o - \frac{\gamma p_o}{L_o}\left(u_o cos(\omega_n t) + \frac{\dot{u}_o}{\omega_n} sin(\omega_n t)\right) \tag{49}$$

The numerical solution to the governing equation is shown in Figure 59, using a temporally converged time step. The plots in this figure are presented to provide the reader with a clear picture of the dynamics of the linear problem.




(a) (b)

Figure 59: Displacement and pressure histories for the simulation of the 0-D linear piston.

Numerical analysis of the simulations is presented in Fig. 60, where the time averaged $L_1$ error is reported. Several time steps are shown, using five different numerical approaches. All simulations used the NASCART-GT built-in mass-spring system solver, except for the single LS-DYNA simulation. The forcing function was also modified to use the linearized pressure of Eq. [43], rather than the aerodynamic loads. The single LS-DYNA simulation applied the linearized forcing function by using an input file specified equation.

Since the solution to the fluid dynamics is an exact solution, and the numerical computation of the mass-spring system is of $2^{nd}$ order accuracy, the coupling scheme can provide, at best, a $2^{nd}$ order accurate solution. The CSS coupling method reduces the global order of accuracy to one, as seen in Fig. 60. Despite the apparent increase in order of accuracy for larger time steps, this is merely due to the extremely large error in this region. As the time steps continue to drop, the formal order of accuracy asymptotes to a value of one. The tightly coupled method performs sub-iterations until convergence is achieved between the displacement and force at each time step. This method achieves $2^{nd}$ order accuracy. The sub-iteration convergence criterion was set such that the force residual fell below a threshold of $1.0 \times 10^{-10} N$. For this specific simulation, sub-iteration convergence was achieved within 5 cycles, with the exception of five time steps that achieved sub-iteration convergence in four steps. Next, the LS-DYNA results are nearly identical to the tightly coupled NASCART-GT simulations. This is expected since the numerical algorithms are the same for both

cases. Finally, the two $2^{nd}$ order accurate coupling schemes (ISS1 and ISS2) demonstrate the expected order of accuracy. For this particular problem, the ISS2 method produces an error that is lower than that of the ISS1 method.

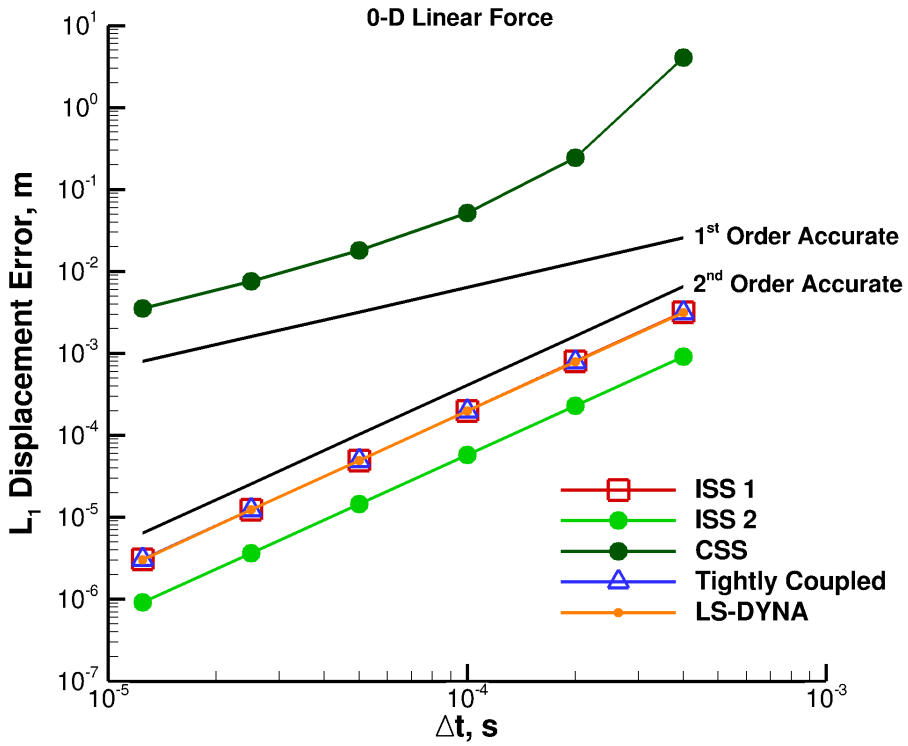
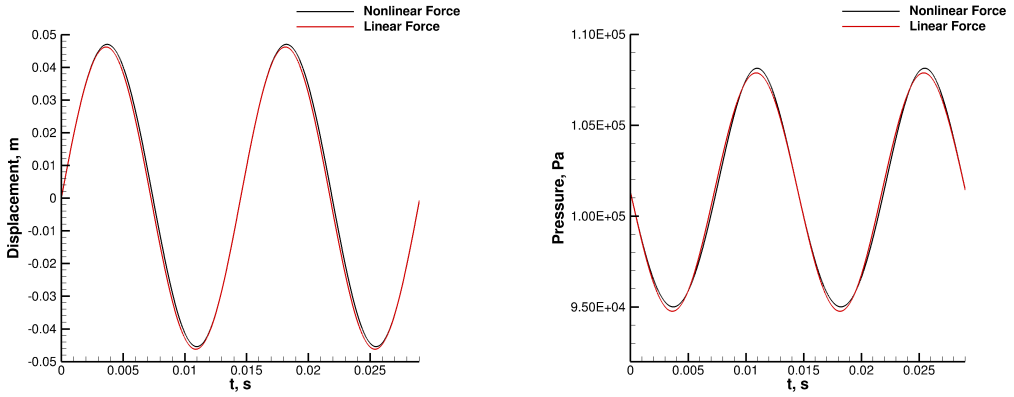



Figure 60: $L_1$ displacement error for the simulations of the linear forcing function.

Next, the same analysis was performed using the nonlinear forcing function. Figure 61 shows the displacement and pressure histories over a small time interval in order to emphasize the slight differences in the solution to the linear force vs. nonlinear force.




(a) (b)

Figure 61: Displacement and pressure histories for the simulation of the 0-D piston using the linear and nonlinear forcing functions.

Once again the $L_1$ error has been plotted as a function of time step for the same methods. Since there is no analytic solution to compute the error against, the errors have been computed using a time step converged simulation. For this particular case, the "exact" solution was treated as the solution from a tightly coupled simulation with a time step of $\Delta t = 1 \times 10^{-6}$. The qualitative results in Fig. 62 are very similar to those from the linear force in Fig. 60. The same conclusions can be drawn, in that the ISS2 coupling algorithm produces the lowest error, and all schemes other than the CSS method, yield $2^{nd}$ order accurate results. Additionally, it is important to note that the largest time step used in all of these analyses led to an unstable simulation using the CSS method, and the simulation failed to complete. This limit demonstrates the important characteristic of improved numerical stability when using the higher order coupling schemes.

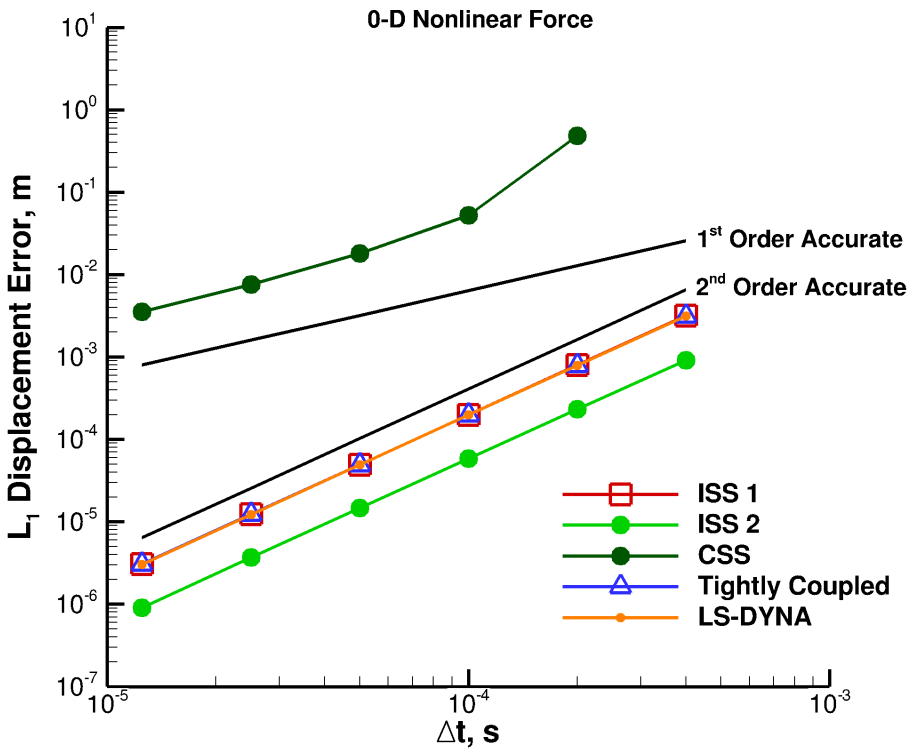



Figure 62: $L_1$ displacement error for the simulations of the non-linear forcing function.

### 5.1.2 Elastic Piston with 1-D Fluid Model

With the validations presented for the 0-D fluid model, the next step is to introduce spatial variations in flow properties through the use of the Euler solver. The fluid dynamics is solved on a 1-D domain using a uniform mesh with 256 computational cells spanning the 1 $m$ length of the fluid chamber. As the piston oscillates about the initial position, the total number of cells vary as cells are consumed by the moving piston, and in the opposite case, emerging from inside the piston.

As presented in the thesis by Piperno [90], the coupled eigenfrequency for the problem can be computed from the fundamental equation as

$$\left(\frac{\omega L}{a}\right) tan\left(\frac{\omega L}{a}\right)\left(1-\frac{k}{m\omega^2}\right)=\frac{\rho_0 L}{m} \tag{50}$$

which results in a fundamental coupled eigenfrequency of 343.4 rad/s, or an oscillation period of 0.0183 s.

Several simulations are presented for this problem, where the fluid dynamics is coupled to both an internal NASCART-GT mass spring solver, as well as externally to LS-DYNA. The coupling within NASCART-GT allows the use of tightly coupled simulations, where the sub-iterations are performed at each time step. These are not possible when coupled with LS-DYNA, as there is no method to cycle back and forth within a given time step until convergence is achieved. In both approaches, the CSS and ISS1 coupling schemes are presented.

Figure 63 plots the displacement history for two simulations against the reference results of Blom [9], and also compares against the analytic expression for eigenfrequency by indicating the width of each period. The simulation titled, "NASCART-GT" is one that uses the internal mass spring solver, while "NASCART-GT + LS-DYNA" is coupled to LS-DYNA. Both results used the ISS1 coupling scheme with a time step of $\Delta t = 2.5 \times 10^{-6} s$. The displacement history for each time step shows very small differences when compared in a single plot. Thus, these two simulations are presented as representative results to demonstrate the physical behavior of the solution. The two results presented in this plot are indistinguishable from one another. These results, however, illustrate some differences compared to the reference. First, the reference results are clearly more dissipative, as the displacement amplitude reduces significantly during the simulation. Despite the differences in amplitude, the frequency agrees very well qualitatively. Likewise, all simulations demonstrate good agreement with the analytic eigenfrequency.

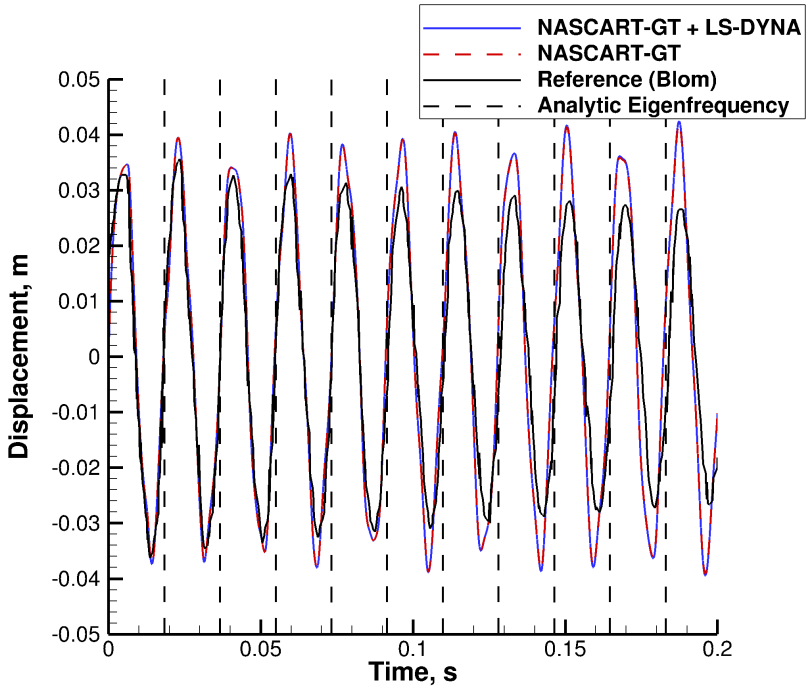



Figure 63: Displacement history for the 1-D elastic piston. The ISS1 coupling scheme with $\Delta t = 2.5 \times 10^{-6} s$ is used for both simulations. Solutions are compared against reference computational data [9] and against analytic eigenfrequency

The $L_1$ error is shown in Fig. 64. The numerical error was computed as the difference between the simulation data at the given time step and the results from a tightly coupled simulation using a time step of $\Delta t = 1 \times 10^{-7} s$. The time steps studied in the present analysis range from $\Delta t = 2.5 \times 10^{-6} s$ to $\Delta t = 2 \times 10^{-5} s$. For all results, the observed numerical error adheres to $1^{st}$ order accuracy. This is unlike the 0-D piston simulations, where $2^{nd}$ order accuracy was achievable using the ISS1 scheme as a result of at least $2^{nd}$ order accuracy in all portions of the analysis. As with the 1-D prescribed motion simulations, the numerical error in the NASCART-GT flow simulations generate numerical error that is heavily dominated by the spatial resolution. That is, the numerical error associated with the spatial discretization is significantly larger than the numerical error introduced by the temporal discretization. The absolute error introduced by spatial resolution can be reduced by refining the mesh. However, due to stability constraints, this requires a reduction in the time step as well. Thus, the time step is never allowed to be large enough for temporal error

to dominate. The conclusion from these observations is that total error will not observe the temporal order of accuracy, and the fluid dynamics will produce at best $1^{st}$ order accurate results.

This is what is observed in Fig. 64, where all coupling schemes demonstrate $1^{st}$ order accuracy. It is observed that the error produced by simulations coupled to LS-DYNA generate larger error, when computed against the temporally converged simulation using the NASCART-GT mass spring solver. These errors are likely due to the fact that the numerical algorithms in both structural solvers are not identical, and will thus produce slightly different solutions.

The two simulations using $\Delta t = 2 \times 10^{-5} s$ were numerically unstable when using the same time step for the fluid and structural solvers. Thus, subcycling was used in order to keep the time step of the CFD to half this value. At the larger time step, the CFD simulation was unstable (i.e. the CFL number was too large). This approach demonstrates the efficient use of subcycling when time constraints differ between disciplines.

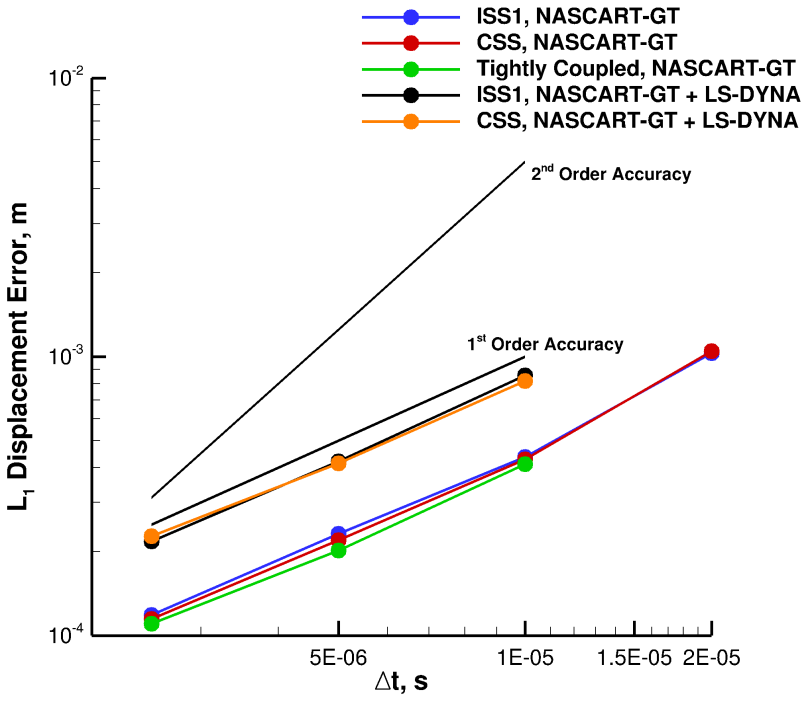



Figure 64: $L_1$ displacement error for the 1-D elastic piston.

## *5.2 Vortex-Induced Vibrations of an Elastic Cylinder*

A 2-D, laminar simulation is considered next, in which the motion of an elastic cylinder is driven by the alternating shed vortices in the flowfield. The model problem is constructed such that flow is computed around a cylinder, attached to two linear springs with one in each coordinate direction. Fig. 65 shows a depiction of this model.

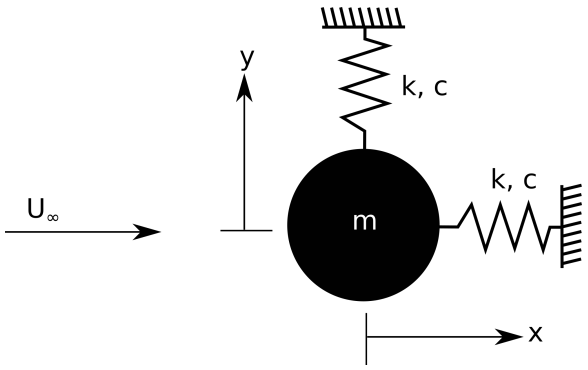



Figure 65: Diagram depicting cylinder of mass, $m$, attached to springs in each coordinate direction. The spring stiffness, $k$, and the damping constant, $c$, are the same for both springs.

The physical properties of this simulation can be characterized by the Reynolds number ($Re = \frac{\rho_\infty U_\infty D}{\mu_\infty}$), Strouhal number ($St = \frac{f_s D}{U_\infty}$), the mass ratio ($M = \frac{m}{\rho_\infty D^2}$), and the reduced damping parameter ($Sg = 8\pi^2 St^2 \alpha M$). Here, $D$ is the cylinder diameter, $f_s$ is the vortex shedding frequency, $m$ is the cylinder mass per unit length, and $\alpha$ is the damping factor in the structural dynamics equations ($\alpha = \frac{c}{2\omega_n m}$ with respect to Eq. [38]). The natural frequency of the cylinder ($\omega_n$) is treated as a parameter that can be varied with respect to the vortex shedding frequency.

This problem has been studied in the past [118, 46, 85], typically varying parameters such as mass ratio, natural frequency, and Reynolds number. The current study presents a single flow condition and a set of structural properties, focused on validating the current computational framework for two dimensional flows. The flow is incompressible, and is modeled as Mach 0.1 flow. The Reynolds number is chosen to be 200, which produces alternating shed vortices that drive the cylinder dynamics. The final parameters have been

chosen based on past computational results [118]: $M^* = 1$, $S_g = 0.01$, $\frac{f_n}{f_s^*} = 1.3$, where $()^*$ quantities are taken from the static cylinder simulation.

### 5.2.1 Static Cylinder

The first simulation considered for this problem is flow past a static cylinder. This serves as a validation, and provides the numerical calculation for Strouhal number. The simulation used Roe's approximate Riemann solver with a $2^{nd}$ order accurate MUSCL extrapolation scheme for the inviscid fluxes. The viscous fluxes were computed using a $2^{nd}$ order accurate central difference scheme, and the $3^{rd}$ order accurate, explicit TVD Runge-Kutta time scheme with a constant time step of $1.0\times10^{-5}s$. The mesh was resolved such that the smallest cell size was approximately equal to 0.005D. It must be noted that this mesh resolution is still very coarse when considering the approximate boundary layer thickness for a flat plate ($\delta \approx 0.002m = 0.001D$) at the given Reynolds number.

A snapshot of the flowfield is shown in Fig. 66. The flowfield is colored by velocity magnitude, showing the alternating shed vortex pattern in the wake. Once the simulation entered into periodic motion, the cell count fluctuated around 540,000 cells. A time history of the lift is shown in Fig. 67. After initial transients, a cyclic pattern of alternating shed vortices begins, as evidenced by the periodic nature of the lift coefficient. The non-dimensional frequency of the lift coefficient as represented by the Strouhal number is $St = 0.2016$, which is within 5% of the reported value in the reference [118]. The peak magnitude in lift coefficient is approximately 0.66, which is under-predicting the reference results by approximately 12%. The mean drag coefficient was computed at 1.46, over-predicting the reference computations by 10.6%.

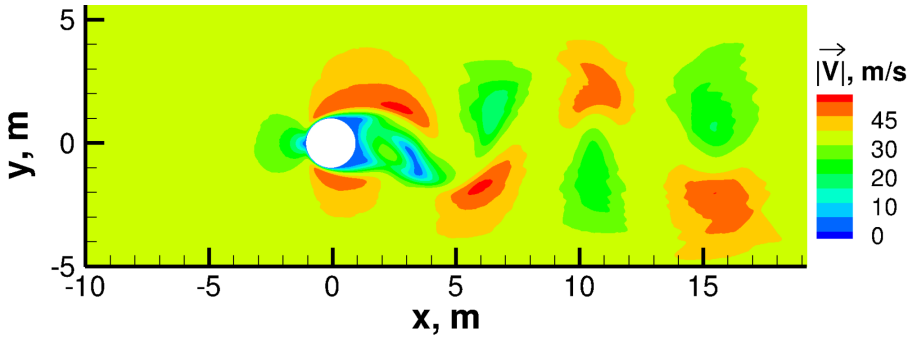



Figure 66: Flowfield for static cylinder, highlighting the vortex pattern in the wake.

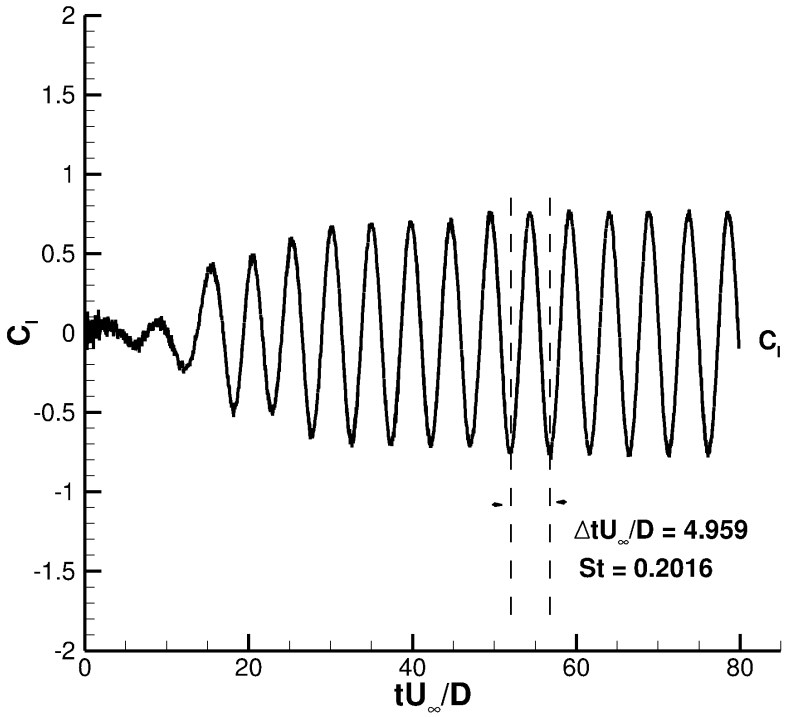


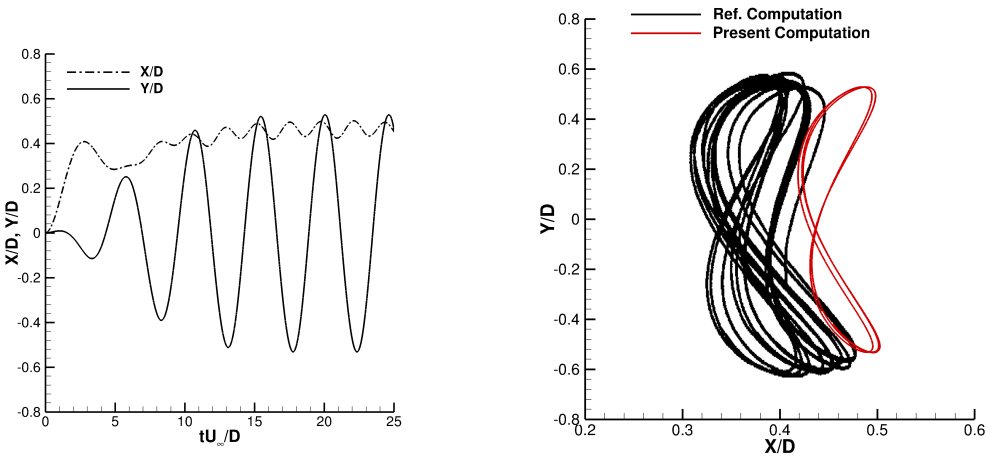

Figure 67: Time history of lift coefficient for the static cylinder simulation.

### 5.2.2 Elastic Cylinder

Next, the elastic cylinder simulation is investigated. Despite the lack of sufficient grid resolution in the static simulation, the mesh for the elastic case required further coarsening. This has been done to decrease simulation time, which has now increased due to the computational cost associated with re-meshing at every numerical iteration. The mesh has been

coarsened by one refinement level, which resulted in a cell size of approximately 0.01D. After sufficient flow time, when the system entered periodic motion, the cell count fluctuated near 100,000 cells.

Time histories of the cylinder position in the x and y coordinate directions are shown in Fig. 68a. The cylinder oscillates in the y-direction about the original position of $y = 0$, as a function of the alternating shed vortices. The cylinder maintains a mean displacement in the x-direction due to the drag generated by the cylinder. The trajectory of the cylinder is shown in phase space in Fig. 68b, where the initial start-up motion has been removed. The present results produce the correct trajectory shape, with close agreement to the peak transverse displacements. The displacement in the axial direction is over-predicted by approximately 25%, which may be attributed to the over-prediction in drag coefficient due to insufficient mesh resolution. Increasing mesh resolution by one refinement level leads to cell counts on the order of 500,000 to 1,000,000. Additionally, the reduction in cell size requires a further reduction in time step, leading to simulations requiring on the order of one to two million iterations. Thus, due to computational time constraints, further investigation into this simulation was terminated. As both the parallel and implicit time stepping capabilities of the current CFD tool advance, future work must continue to validate this viscous FSI problem.




(a) (b)

Figure 68: Displacement time histories and phase plots for elastic cylinder simulation. Comparison made against reference simulation [118].

### *5.3 Shock Wave Impacting a Deforming Panel*

The next validation considers shock waves impacting a deforming panel. This problem has been studied by Giordano et al. [44] and Brehm et al. [12]. The experimental setup is shown in Fig. 69. The inlet is on the left, and a shock tube on the right. The deformable panel is attached to a rigid base structure. There are two panel lengths studied (40 $mm$ and 50 $mm$). A pressure sensor is located near the upper wall and measures the time history of the flowfield pressure.

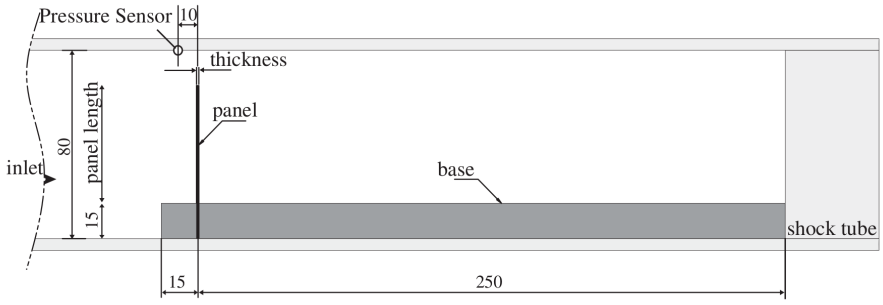



Figure 69: Experimental schematic (Giordano et al. [44]).

### 5.3.1 Fluid Dynamic Analysis of Rigid Configuration

The first set of analyses look at the flow past a rigid panel, with the purpose of determining sufficient mesh resolution. The computational domain is shown in Fig. 70 with the initial conditions listed on the left and right hand side.

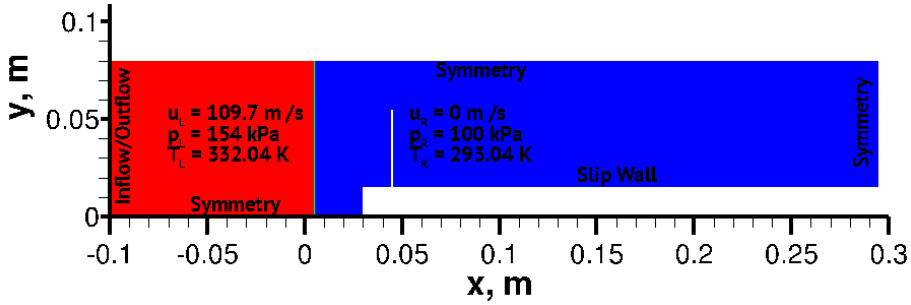


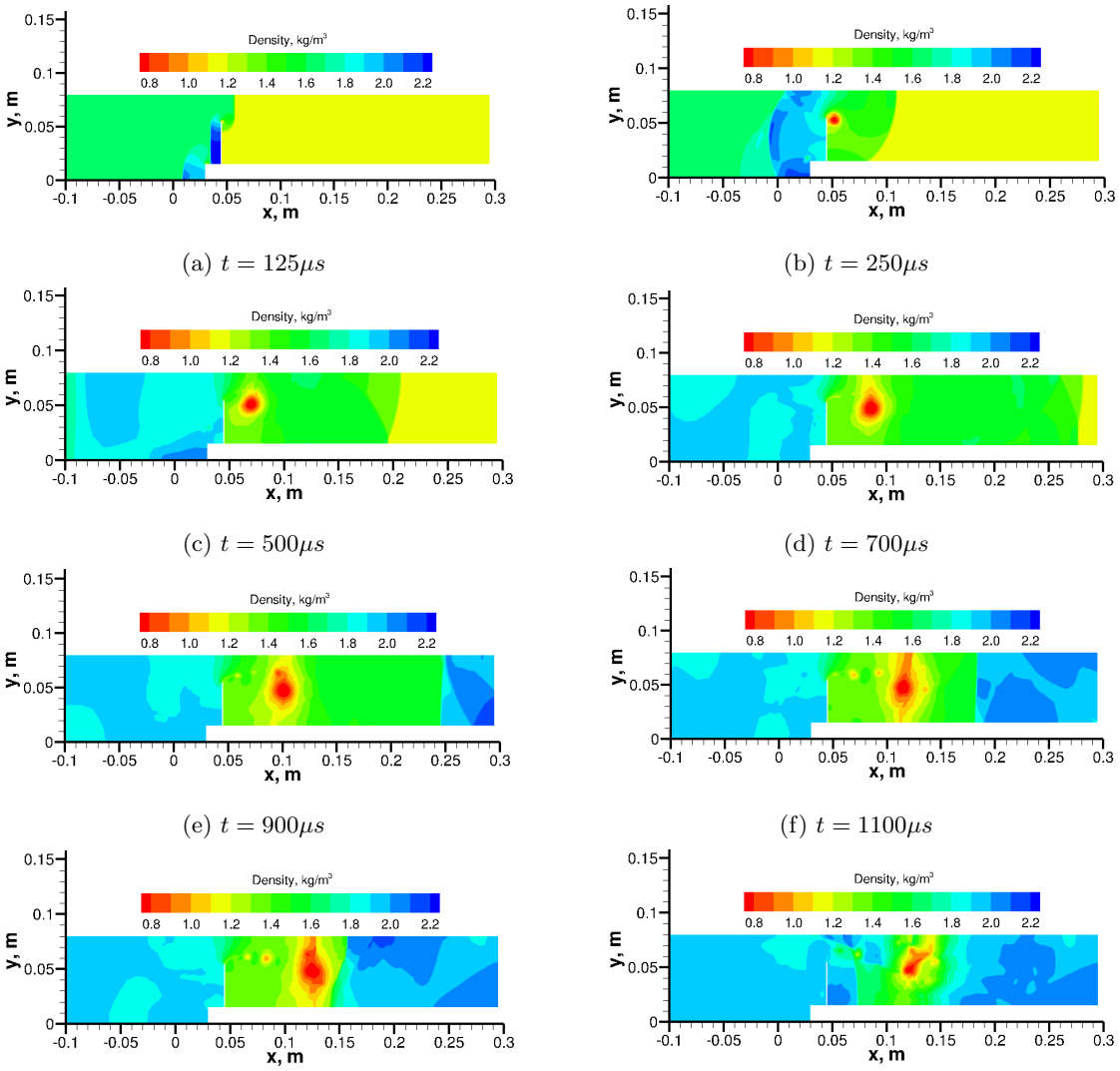

Figure 70: Initial conditions for the left and right side of discontinuity. Flowfield boundary conditions are also listed. Note that the left side of the domain actually extends to an x-value of -0.505 $m$, but not shown here for the sake of clarity.

The panel was given its true physical thickness of 1 $mm$, which puts a minimum cell size requirement on the mesh. A grid convergence study examines three mesh refinement levels. For the purposes of this discussion, the meshes will be described as coarse, medium, and fine. For all simulations, the inviscid fluxes were computed using AUSMPW+ with $2^{nd}$ order accurate MUSCL extrapolation using the MinMod limiter.

The flowfield was initialized into two distinct regions to provide the correct conditions to produce the Mach 1.21 shock wave. The initial location of the shock wave is arbitrary, provided that it is upstream of the base of the structure. The location was specified such that it lined up perfectly along a straight vertical line of cell faces. This provided a well-posed initial condition. The upstream state was set to a velocity of 109.7 $m/s$, a pressure of 154 $kPa$, and a temperature of 332.04 $K$. The downstream state was set to a velocity of 0 $m/s$, a pressure of 1 $kPa$, and a temperature of 293.04 $K$. The base, panel, and upper and lower walls are specified with a slip velocity condition. The inlet is set to use Riemann Invariants, while the right hand side is set to a symmetry plane condition.

Snapshots of the flowfield are shown in Fig. 71, where the flowfield is colored by density.

The shock wave is initiated from ahead of the structure, and the first image is immediately after the shock wave reflects off the base and panel. In Figs. 71b through 71d, the transmitted shock travels through the channel, reflects off the right edge, and travels back towards the panel. During the remaining images, the shock wave interacts with the vortex that was shed from the tip of the panel.




(a) $t = 125\mu s$

(b) $t = 250\mu s$

(c) $t = 500\mu s$

(d) $t = 700\mu s$

(e) $t = 900\mu s$

(f) $t = 1100\mu s$

(g) $t = 1200\mu s$

(h) $t = 1500\mu s$

Figure 71: Rigid panel flowfield showing density contours. Time intervals selected to show key flow features. Solution was computed using the fine uniform mesh.

Mesh convergence is studied by comparing the time history of relevant quantities as a function of time. The first set of simulations studied three different mesh refinement levels.

The coarse mesh contained approximately 7,000 cells, the medium mesh approximately 25,000 cells, and the fine mesh approximately 100,000 cells.

Figure 72 shows the time history of the axial force coefficient of the entire computational surface. The initial rise and fall in axial force is due to the increased load immediately after the shock wave impacts the base of the configuration, as well as the panel. As the shock moves off of the panel, the pressure drops, and then as the shock passes through the upper region of space between the panel and wall, the pressure on the back side of the panel increases. This leads to an overall reduction in axial force. Eventually, as this shock wave reflects off of the right hand wall, and travels back towards the panel, the force undergoes a steep drop upon impact with the panel. From this point onward, the fluctuations are caused by the continuous reflection of waves traveling in either direction.

The pressure history is shown in Fig. 73, where the pressure sensor is located near the upper wall, just upstream of the panel. Thus, the first rise in pressure is due to the initial passing of the shock wave, and then quickly rises again when the shock reflects off of the panel. The reflection of the shock wave off the right hand wall, eventually returns to the pressure sensor, and produces the $2^{nd}$ large rise in pressure. As with the axial force, the pressure also undergoes fluctuations as the reflection of waves continue to propagate throughout the chamber.

In both the force and pressure plots, comparisons are made between the three different mesh refinement levels. The major features, including the strongest instances of shock wave interactions, agree well between all meshes. Differences in the in the oscillatory behavior are evident during other instances in time, and are due to both the cell size, as well as the progression of solution-based mesh adaption. There is no clear trend towards grid convergence when considering any arbitrary instant in time. As the shock wave first approaches the panel and reflects off of it, the three meshes produces indistinguishable results. During the next time interval, the coarse mesh and medium mesh results are in close agreement, with the fine mesh results beginning to deviate. However, the next significant event at $t \approx 1.5 \times 10^{-3} s$, shows nearly perfect agreement between meshes again. As the waves continue to propagate through the domain, the results generally agree well.

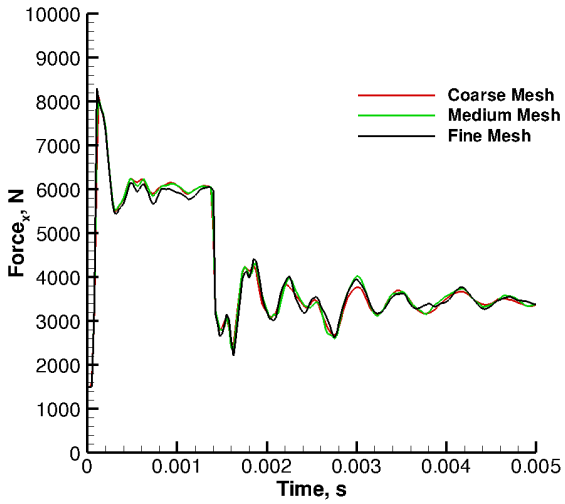



Figure 72: Time history of computational surface loads for three grid refinement levels.

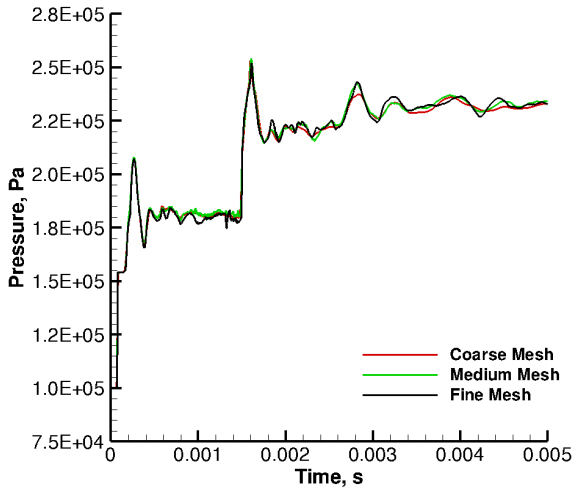
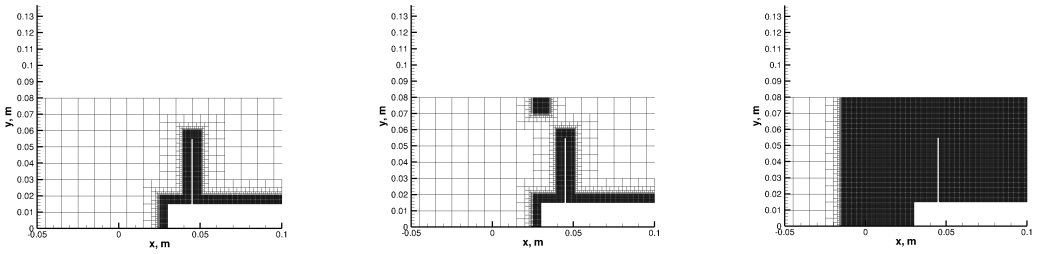



Figure 73: Time history of pressure for three grid refinement levels.

The previous three meshes that were studied simply demonstrated variations in the smallest cell refinement levels. Next, three different meshing approaches are considered, all using the smallest cell size of the fine mesh case. Figure 74 shows the three meshes used. The first approach is the baseline, fine mesh grid that was used in the previous analysis. The surface cells have been refined to the a given refinement level, and solution adaption is used to capture gradients in the flow. The next mesh starts from the baseline case, and adds a small region of refined cells where the pressure sensor is located. This region of cells

was not allowed to undergo any adaption during the simulation. The purpose of adding this region is to ensure that the interpolation of pressure at the sensor location is as accurate as possible within this region of uniform cells. Finally, the third mesh is considered a uniform mesh. The uniform portion of the mesh begins upstream of the initial shock location, and extends throughout the entirety of the downstream portion of the chamber. This simulation does not require any solution adaption.




(a) (b) (c)

Figure 74: Three different initial mesh strategies were considered in the CFD analysis. (a) utilizes the typical meshing strategy with refinement near the surface, and is treated as the baseline case. (b) adds a region of uniformly refined cells in the location of the pressure sensor. (c) is a uniformly refined mesh in the region containing the base and panel.

A comparison of axial force history for these three meshes is shown in Fig. 75. Once again, there is fairly good agreement between these three approaches. The maximum deviations in force are less than 5%. The pressure histories are shown in Fig. 76, with the major features in near perfect agreement. The smaller fluctuations show deviations with low relative error between each other. These observations are useful because they show that it is sufficient to run simulations without the need for extra regions of uniform refinement. The traditional approach of refining at the surface, and adapting to velocity divergence is sufficient for capturing the time history of flow properties for a static beam.

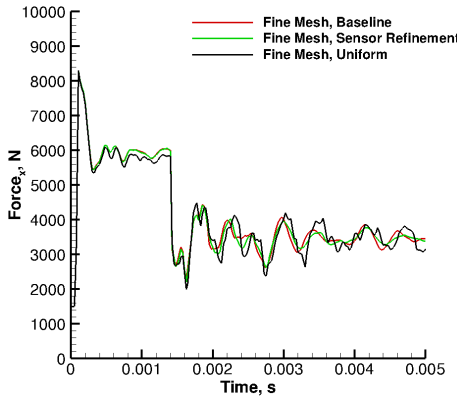



Figure 75: Comparison of computational surface loads using fine mesh for three refinement strategies.

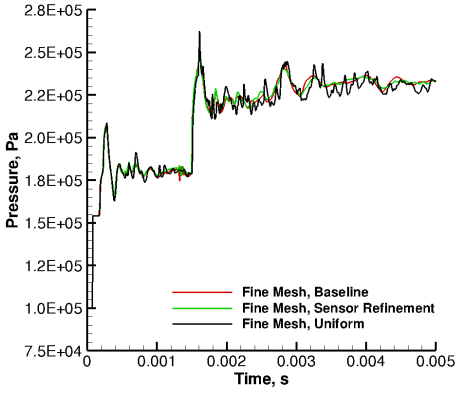



Figure 76: Comparison of pressure history at sensor location using fine mesh for three refinement strategies.

### 5.3.2 Structural Dynamics

The LS-DYNA beam model is used to compute the dynamics of the deforming panel. The Belytschko-Schwer resultant beam element formulation has been used, which requires specification of the cross-sectional area ($A = 1.0 \times 10^{-6}\ m^2$) and moments of inertia ($I_{xx} = I_{yy} = 8.33 \times 10^{-14}\ m^4$, $J = 1.406 \times 10^{-13}\ m^4$). The beam density is set to $\rho = 7600\ kg/m^3$, and the elastic modulus is $E = 220\ GPa$. The LS-DYNA simulations utilize the implicit dynamics approach, where the $2^{nd}$ order accurate Newmark time integration scheme is used. A constant time of $\Delta t = 2.0 \times 10^{-7}$ s is used.

### 5.3.3 Coupled Analysis of Deforming Panel

The FSI simulations all used the CSS coupling algorithm, using the medium mesh approach discussed previously. Two beam lengths have been studied, with the first being the 40mm beam deflection case. Figure 77 shows the time history of pressure at the sensor location. The flowfield is initialized with a discontinuity to the left of the beam and base. Since this location is not in line with the same location as the experiment and reference numerical results, all results must be adjusted such that the physical times correspond with one another. This has been determined by noting the time at which the first pressure rise occurs that is caused by the incident shock wave. The difference in this flow time between the reference data and the current numerical simulation provides the necessary time shift. This can be seen in Fig. 77 where the pressure rise due to the passing of the incident shock wave matches between the present results and the reference data.

Two sets of numerical results using the present approach are shown in Fig. 77. The baseline NASCART-GT simulation used the standard solution adaption approach, based on velocity divergence. The uniform mesh simulation did not require solution adaption since the mesh was set to its finest level across the entire grid. The pressure rise due to the incident and reflected shock waves agree well with the numerical reference data, and even the experimental data, except for the experimental noise immediately after the incident shock wave passes. For the remainder of the simulation, the present numerical results and reference numerical results over-predict the pressure compared to the experimental data. The two NASCART-GT simulations are very similar, with higher frequency modes captured by the uniform mesh.

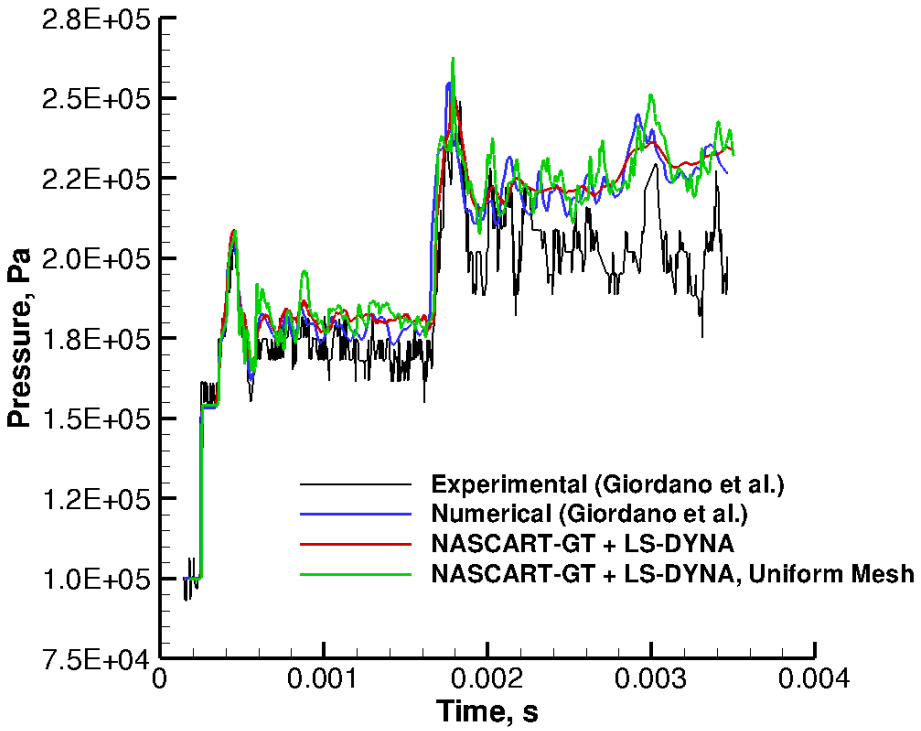



Figure 77: Time history comparison of pressure at the sensor for the 40 mm panel. Comparisons are made against the experimental and numerical results of Giordano et al. [44].

Figure 78 shows the time history of tip displacement of the deforming panel. The present results are within nearly all of the experimental uncertainties, except for some portions of data, where the experimental values appear to lie outside the overall sinusoidal motion. During these portions of the simulation, the NASCART-GT results agree with the reference numerical results, which also adhere to the sinusoidal response.

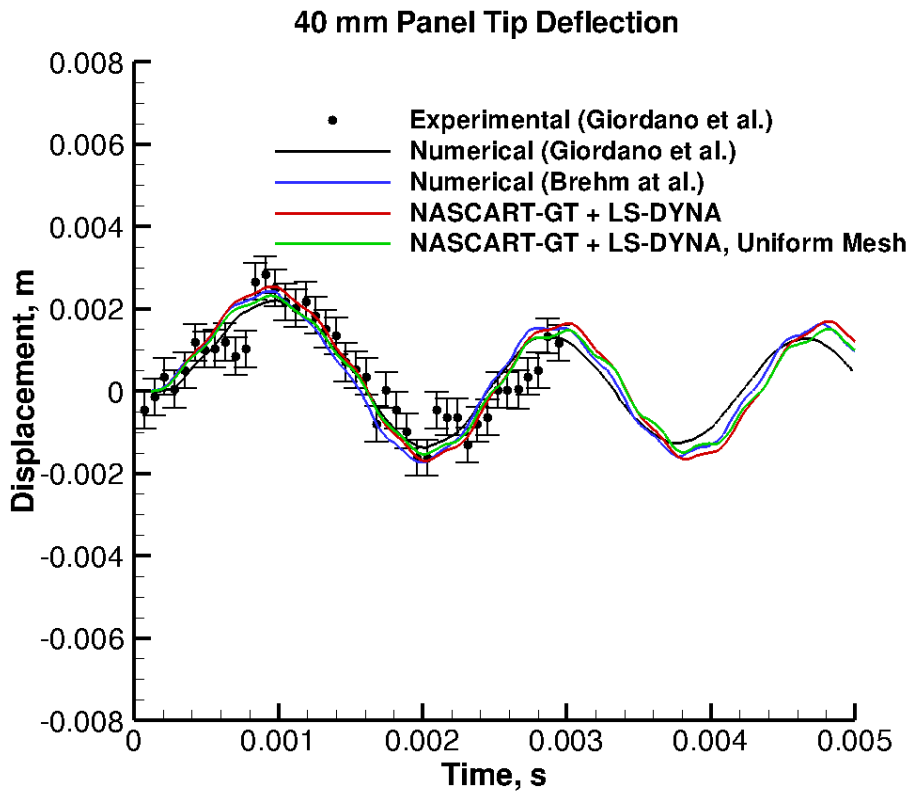



Figure 78: Comparison of present results with the experimental and numerical results of Giordano et al. [44] and Brehm et al. [12]

The 50 *mm* panel is investigated next. The pressure and tip displacement plots are shown in Figs. 79 and 80, respectively. The initial increases in pressure due to the incident and reflected shock waves compare well with the reference numerical results, as well as the experimental data. Beyond this point in the simulation, deviations in pressure are seen when compared against the reference numerical results, as well as the experimental data. The displacement plot also shows the same qualitative trends. Initially, all results presented in Fig. 80 are in close agreement. However, the numerical results begin to deviate from the experimental results very quickly. This was discussed in Ref. [44], where the authors noted that the base of the configuration, where the panel is attached, appears to deform. This displacement contributes to the overall motion, which in turn affects the coupling of the problem. The NASCART-GT results over-predict the magnitude of the displacement in the first period, compared to the experimental data and results of Ref. [44]. However, this

first peak agrees well with the results of Ref. [12]. Beyond this point in time, all numerical results presented in this figure begin to deviate. It should be noted that smaller time steps were studied with negligible effects on the solution. Despite the differences in these results, they still show a stable simulation with accurate predictions of the maximum displacements when compared to the reference numerical results. Though it is not within the scope of this thesis, further investigation into modeling the base motion observed in the experiment would be of interest.

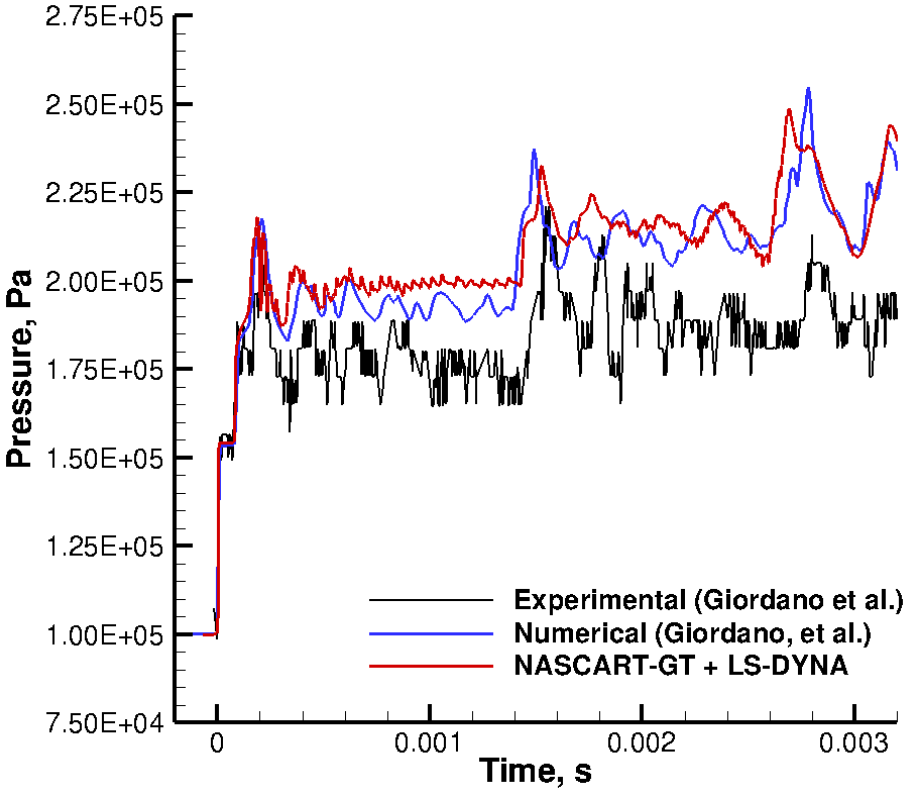



Figure 79: Comparison of time history of pressure at sensor for 50 mm panel against the experimental and numerical results of Giordano et al. [44]

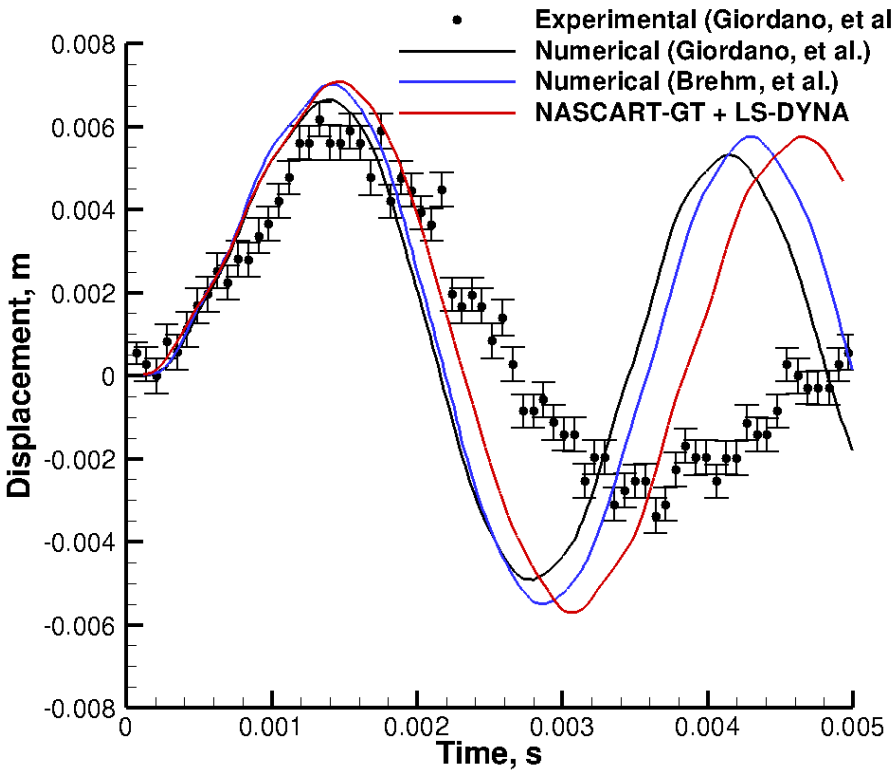
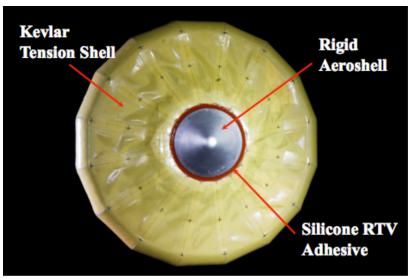



Figure 80: Comparison of present results for 50 mm panel with the experimental and numerical results of Giordano et al. [44] and Brehm et al. [12]

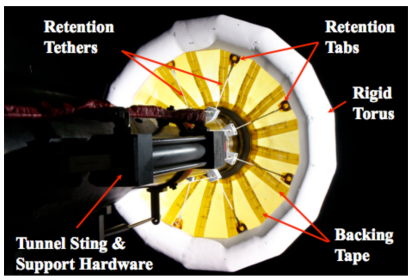
### *5.4 Steady-State FSI Analysis of Semi-Rigid Tension Cone*

A 3-D configuration of interest is the semi-rigid tension cone, which has been studied experimentally and computationally. The experimental results performed at the NASA Glenn $10 \times 10$ ft Supersonic Wind Tunnel are documented by Clark [22], and are used as validation data in the present study. Additionally, this problem has also been studied from a computational standpoint by Tanner [107], which will also be used as a means of validation. Detailed descriptions of the model can be found in the paper by Clark, et al. [22] and the PhD thesis by Tanner [107]. The configuration is shown in Figure 81, where photographs of each side of the tension cone are shown. From the windward perspective, the rigid aeroshell is identified in the center, with the Kevlar tension shell deployed around the model capsule. From the leeward side, the rigid torus is made visible, as well as the wind tunnel sting. The modeling details of the configuration will be explored further in the CFD and FEA

analyses respectively. In this section, independent analyses for the CFD and FEA are examined first. Then, a discussion on the mesh mapping and data transfer follows. Finally, the loosely coupled, steady state analysis is presented.




(a) Windward side of the semi-rigid tension cone.

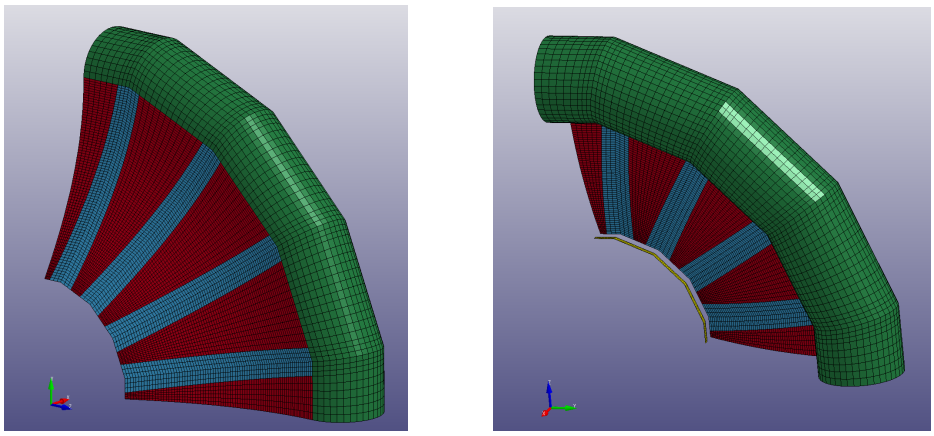



(b) Leeward side of the semi-rigid tension cone.

Figure 81: Images of the experimental semi-rigid tension cone, as shown in the reference work by Clark [22].

### 5.4.1 Structural Analysis

First, an FEA simulation of the tension cone is presented to discuss the computational modeling approach. The use of words "part", "material", "section", and "set" in this section will be done so in the context of the associated LS-DYNA keywords. The LS-DYNA mesh representing one quarter of the geometry is shown in Fig. 82. The tension cone model is comprised of three components, further broken down into four parts. The components are identified as the tension shell, the torus, and the sting connector. The tension shell is split into two LS-DYNA parts, which include the primary tension shell (identified as red in the figure), and the seams used to hold the fabric together (identified as blue in the figure). The torus, identified in green, and the sting connector, identified in yellow, are both treated as rigid bodies, undergoing no material deformations.



(a) Windward side.

(b) Leeward side.

Figure 82: The LS-DYNA FEA mesh is shown with parts indicated by different colors. The tension shell is broken into two parts: the primary shell (red) and the seams (blue). The torus is green and the sting connector is yellow.

The tension shell is made of a urethane-coated Kevlar, where the taped seams are modeled by doubling the thickness (as was done by Tanner [107]). The tension shell is separated into two parts, so different section properties can be associated with the seam and non-seam portions. It is the section properties that specify the thickness and element formulation. The tension shell material is modeled as an isotropic fabric. Both the torus and the sting connector are modeled as rigid materials with translational and rotational constraints. For the simulation of the quarter geometry with symmetry planes about the Y and Z axes, the torus is only allowed to translate in the axial direction with no rotational motions permitted. The sting connector is constrained in all degrees of freedom, and is not needed for the structural analysis. However, it was found to be a useful addition to the configuration during the mesh data transfer steps, and this will be discussed in more detail in that section.

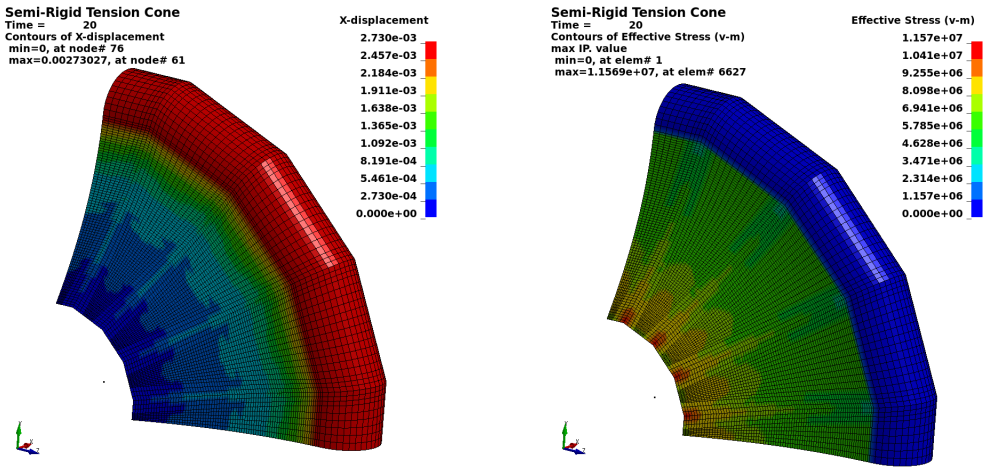
Three sets of boundary conditions are used for this model. Symmetry conditions are set along the Y and Z symmetry planes. Note that there are no constraints on the symmetry plane nodes that are part of the rigid bodies. This is not required since the entire rigid body is already constrained to the allowable motions. The nodes along the lower edge of

the tension shell are set to be constrained in all degrees of freedom since this is where the tension shell is attached to the aeroshell. Additionally, a contact surface has been identified to include the possible interaction between the tension shell and the torus. Due to the fact that these are all steady state cases, implicit equilibrium calculations are carried out. The model contains a total of 8,936 nodes and 8,656 shell elements. Table 5 lists the material properties for the tension cone.

Table 5: Material properties for the semi-rigid tension cone.

| Material Properties | |
|---|---|
| E | 6.6593 GPa |
| $\rho$ | 1125.86 kg/m$^3$ |
| h | 3.3401 x $10^{-4}$ m |
| $\nu$ | 0.3 |

The applied loads for this simulation are those derived from the aerodynamic loads of a static CFD simulation. This serves as a means of verification that the loads produce expected results. Figure 83 shows the axial displacement and the stress distribution on the converged simulations. The maximum axial displacement is reported as $2.73 \times 10^{-3} m$. The displacement contours show variations along the seams, where the material thickness increases. The effects of these regions can also be seen in the stress distribution, where there are local maxima where the seams attach to the aeroshell.



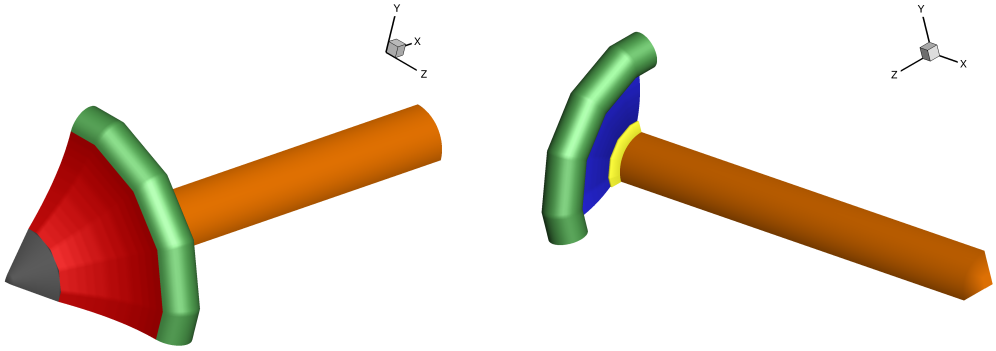

(a) Axial displacement (m).

(b) Von mises stress (Pa).

Figure 83: Steady state solutions with an applied aerodynamic load.

### 5.4.2 Fluid Dynamic Analysis

The corresponding CFD model is now explored for the simulation of the tension cone. The entire configuration must now be considered, including the aeroshell and wind tunnel sting, in order to properly capture the flow physics. The model consists of a total of six separate components that are identified in Figs. 84 and 85: 1) aeroshell, 2) front portion of tension shell, 3) torus, 4) back portion of tension shell, 5) sting connector plate, and 6) wind tunnel sting. Each of the components is treated as an independent input STL geometry within NASCART-GT. There are a total of $58,960$ elements making up the CFD input geometry configuration. The distribution of elements are most critical on the surfaces where pressure data is transferred to the FEA mesh. Thus, the aeroshell and sting models have been created with sufficient mesh resolution required to represent the CFD input geometries from a surface representation standpoint, but not necessarily from a pressure mapping standpoint. This allowed for a significant reduction in the number of elements, especially along the sting, where axial variations in surface properties are not necessary.

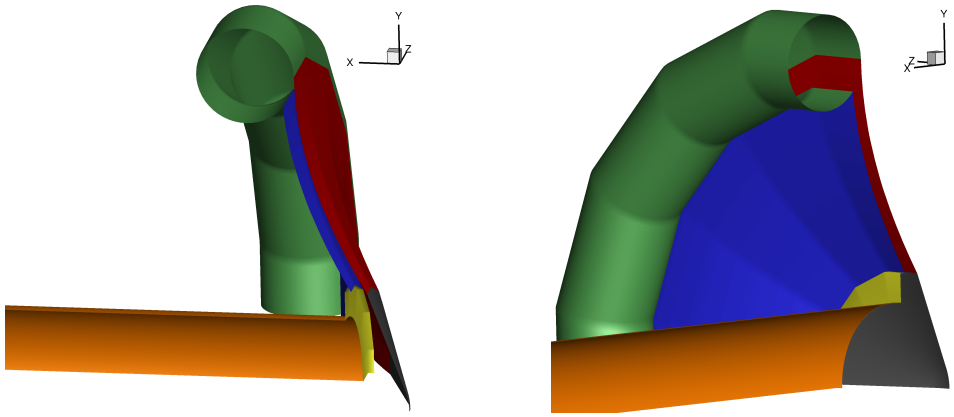



(a) Isometric view from the front.

(b) Isometric view from the rear.

Figure 84: CFD input geometry configuration consists of six components: 1. aeroshell (gray), 2. tension shell (red), 3. torus (green), 4. rear tension shell (blue), 5. sting connector (yellow), 6. sting (orange).

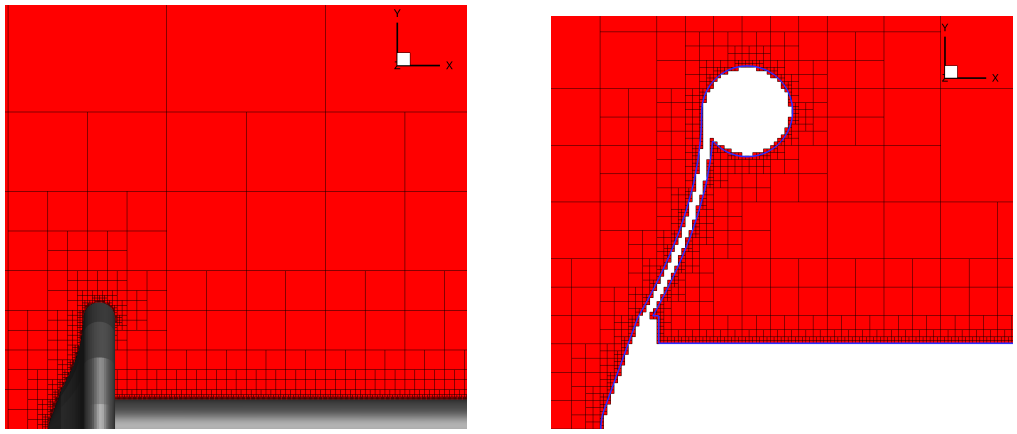
Figure 85 highlights an important difference between the CFD and FEA input geometries. One of the requirements in generating the CFD volume mesh is that the input geometry forms a closed surface with a distinct inside and outside region. Using the FEA model does not satisfy this condition, since the tension shell is a single fabric, with a thickness much smaller than the CFD spatial scales. An artificial thickness has been added to the input geometry, with the thickness governed by the computational cell size requirements. A smaller thickness requires smaller computational cells to resolve, which leads to very large cell counts, and computationally expensive simulations. On the other hand, a larger thickness leads to a degradation in model accuracy as the CFD and FEA meshes become increasingly different. Thus, a compromise must be made between simulation accuracy and computational cost. This modification has been achieved by removing a small portion of the torus, and creating a secondary tension shell that sits behind the primary tension shell, and connects directly to the sting connector.



(a) (b)

Figure 85: CFD input geometry as viewed from the negative Z side of the symmetry plane.

An example of the initial mesh is shown in Fig. 86, where the symmetry plane is shown. The details of the mesh near the surface are shown in both figures. The initial mesh contained $93,334$ computational cells.



(a) (b)

Figure 86: The initial mesh around the CFD model is shown with (a) depicting the 3-D model and (b) highlighting the mesh near the surface along the Z-symmetry plane.

A CFD simulation is presented for the purposes of characterizing the flowfield, and determining optimal input parameters for this problem. The freestream conditions are listed in Table 6. The quarter geometry has been used for this simulation since the angle of attack is zero, and there is a flowfield symmetry about the X-Y plane and X-Z plane.

Table 6: Flow conditions for the semi-rigid tension cone simulation.

| Freestream Conditions | |
|---|---|
| $M_\infty$ | 2.441 |
| $\rho_\infty$ | 2.6776 x $10^{-2}$ kg/m$^3$ |
| $p_\infty$ | 1034 Pa |
| $\alpha$ | $0^o$ |

An inviscid simulation was run with slip wall boundary conditions on the surface of the tension cone. Farfield boundary conditions (Riemann invariants) were applied to all domain boundaries, except for the symmetry planes. The simulation was run for $6,000$ iterations, where the mesh was adapted every 100 iterations. The time scheme was set to the $3^{rd}$ order accurate TVD Runge-Kutta scheme with a constant CFL number of 1.0. The AUSMPW+ inviscid flux scheme was used with the $1^{st}$ order accurate unstructured gradient calculations. Solution adaption was used based on velocity divergence, with an extra constraint on the minimum cell size for adapted cells. In order to reduce the cell count, and computational cost, the minimum adaption cell size was set to be of one refinement level smaller than the surface cells. This approach provided sufficient mesh resolution along the shock, while still resolving the input geometry to the desired accuracy, and kept the final cell count down to $714,615$. Figure 87 shows the final result with symmetry plane slices colored by Mach number, and the tension cone geometry displaying surface pressure coefficient, $c_p$.

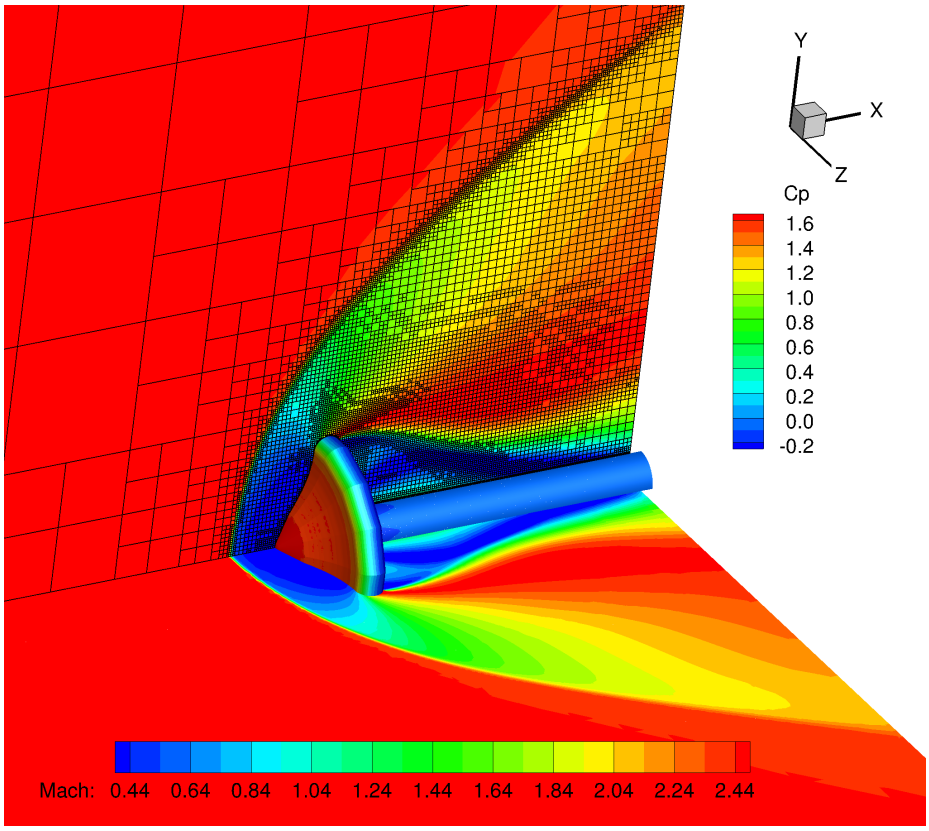



Figure 87: Steady state CFD solution. The flowfield is colored by Mach number, while the surface pressure coefficient is shown on the surface.

### 5.4.3 Mesh Mapping and Data Transfer

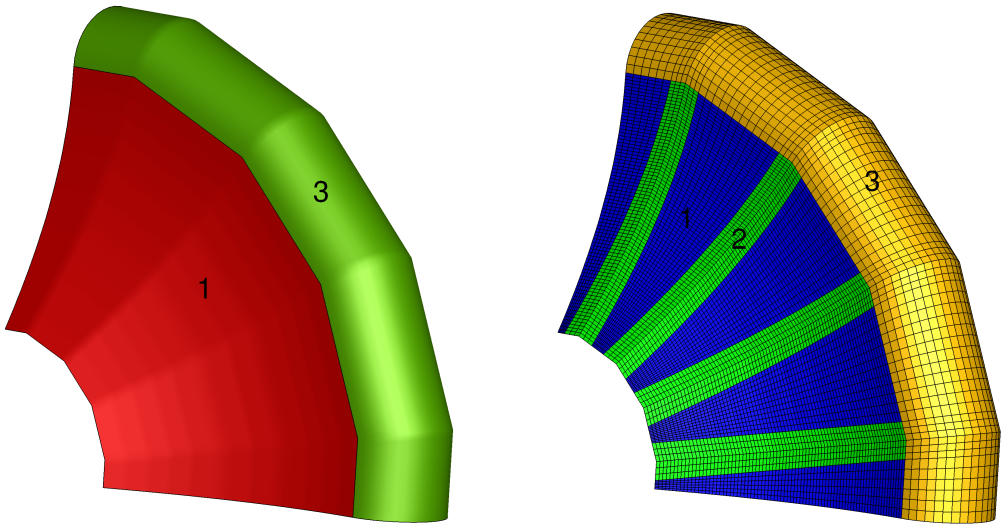
The mapping of surface data between the FEA and CFD meshes has been done using DDTBM. This section describes the details of this process with respect to the steady state FSI analysis of the semi-rigid tension cone.

The front portion of the CFD tension shell mesh and the CFD torus mesh are nearly identical to their respective FEA meshes. Since NASCART-GT requires an input geometry of triangular elements, the quadrilateral elements of the FEA mesh were split in half. The extra, artificial tension shell in the CFD model is topologically similar to the front potion, but independently generated. Figures 88 and 89 illustrate the two configurations. Note that the CFD mesh only requires the parts for which data mapping is necessary (i.e. the aeroshell and sting components are not needed during this process). However, for the FEA mesh, the addition of a sting connection was found to be necessary. The purpose of this

component is to ensure that the artificial CFD tension shell remained attached to the sting at all times.

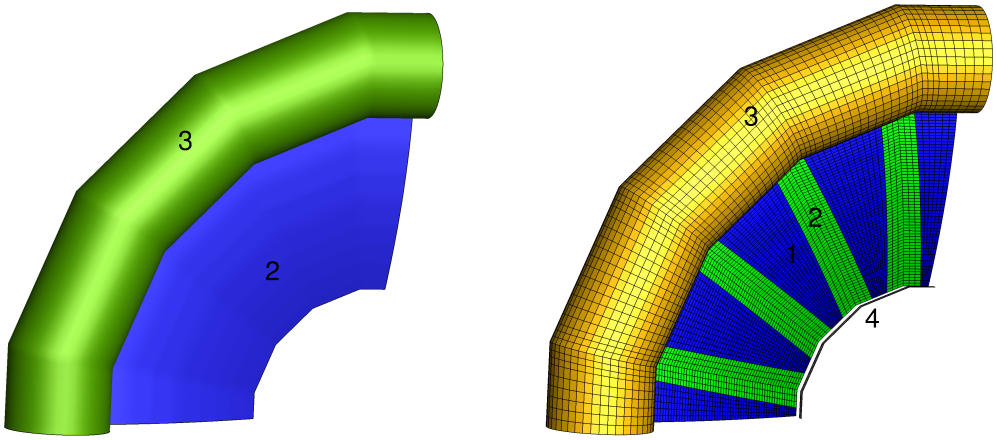
Data mapping is performed in two directions: 1) from the CFD mesh to the FEA mesh and 2) from the FEA mesh to the CFD mesh. In the first scenario, there are two mappings that are identified. The first mapping requires that the aerodynamics of CFD parts 1 and 2 (both tension shells) are mapped to FEA parts 1 and 2 (tension shell non-seam part and tension shell seam part). That is, the pressure field on the front and back sides of the CFD tension shell is transferred to the entire FEA tension shell. The second mapping corresponds to the torus and is accomplished by simply mapping the CFD part 3 (torus) to FEA part 3 (torus). Once the pressure fields are interpolated from the CFD mesh to the FEA mesh, the pressures are converted to nodal forces, which can be used as input to LS-DYNA.

Now, moving in the opposite direction, three mappings have been identified. In this direction, it is the displacement field on the FEA mesh that is interpolated to the CFD mesh. The first mapping consists of FEA parts 1 and 2 (entire tension shell) being mapped to CFD part 1 (front tension shell). Note that the displacements on the FEA mesh have not been directly mapped to the artificial CFD tension shell. It was found that this approach did not adequately constrain the artificial tension shell to stay in contact with the sting connector. It is for this reason that the sting connector was added to the FEA mesh, even though is it not relevant in the FEA calculation. Thus, the second mapping transfers data from FEA parts 1, 2, and 4 (tension shell non-seam, tension shell seam, and sting connector) to CFD part 2 (artificial tension shell). The third mapping is between FEA part 3 (torus) and CFD part 3 (torus).




(a) (b)

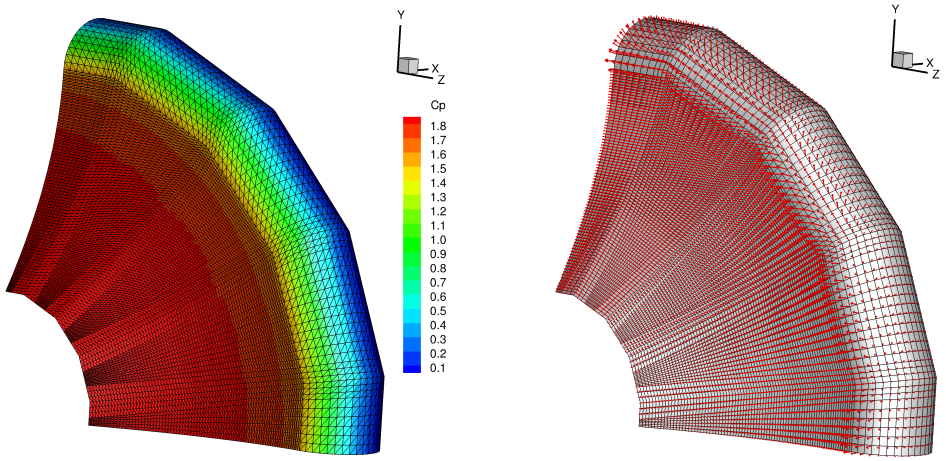
Figure 88: Front view of CFD and FEA meshes with part identification numbers used for data mapping.




(a) (b)

Figure 89: Back view of CFD and FEA meshes with part identification numbers used for data mapping.

The aerodynamic load transfer to the FEA mesh is depicted in Fig. 90. The load transfer consists of two internal processes taking place. First, the surface pressure distribution on the CFD mesh is interpolated to the FEA mesh. Then the surface pressure on the FEA mesh in converted into a nodal force distribution.

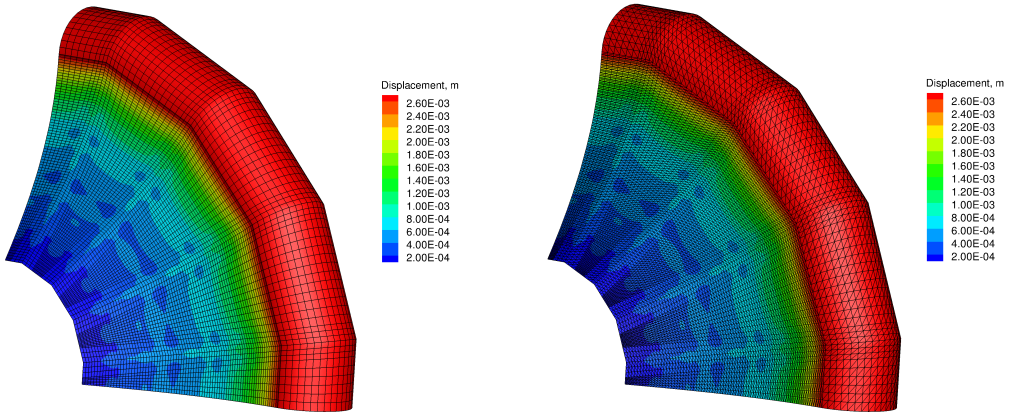



(a) CFD aerodynamic loads.

(b) FEA nodal forces.

Figure 90: Aerodynamic loads are shown in the form of surface pressure coefficient, with the corresponding nodal force vectors on the FEA mesh.

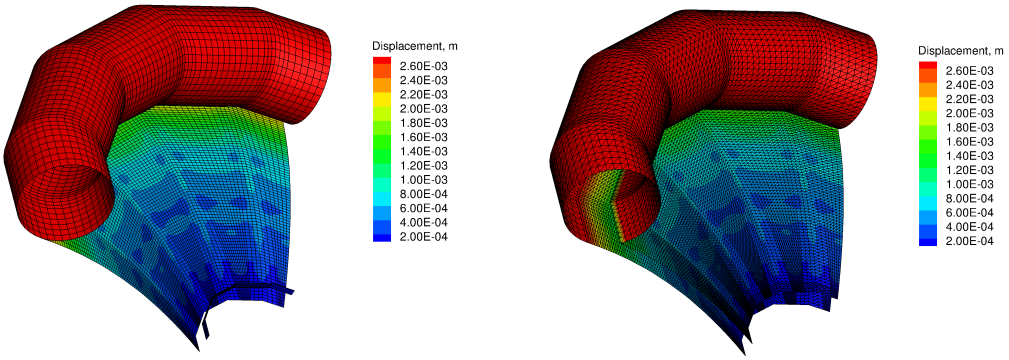
The reverse process, which takes the nodal displacement on the FEA mesh to the CFD mesh is shown in Figs. 91 and 92. The displacement distributions, qualitatively look very similar, as expected. Fig. 92 is especially useful in providing further understanding into how the artificial CFD tension shell is mapped to the FEA mesh. The displacement distribution on the back portion of the CFD tension shell mimics the displacements from the FEA tension shell, as was the objective. It can also be seen that the artificial CFD tension shell remains in contact with the torus, and although not shown, remains in contact with the sting connector.




(a) Front view of FEA displacement.

(b) Front view of CFD displacement.

Figure 91: Displacement distributions are shown on the front side of the FEA and CFD meshes.




(a) Back view of FEA displacement.

(b) Back view of CFD displacement.

Figure 92: Displacement distributions are shown on the back side of the FEA and CFD meshes.

Finally, a discussion on the use of symmetry planes is necessary. For the configuration discussed thus far, there have been two symmetry planes. These symmetry planes are correctly enforced within the FEA model, and nodes along the symmetry plane always remain along the symmetry plane with zero displacements. The CFD grid is designed such that the edge of the grid is aligned with these symmetry planes. Thus, if the geometry becomes misaligned with the mesh along the symmetry planes, numerical problems could arise. Indeed, it was found that this can happen as a result of the data transfer process.

Mapping displacements from the FEA mesh to the CFD mesh can result in symmetry plane nodal displacement that are non-zero, but near machine zero. If these displacements are large enough, then there would be accuracy concerns. However, due to their near-zero values, the concern is less focused on accuracy and more focused on a negative side effect encountered during the automated CFD meshing. These near-zero displacements could lead to gaps between the CFD input geometry and the edge of the CFD grid, which has the unintended consequence of flood filling the mesh inside the geometry. A simple solution to overcome this was developed. The symmetry planes are identified within the CFD input motion file. When the updated nodal displacements are read back into the CFD simulation, the symmetry plane nodes are overwritten such that displacements are properly constrained by the intended symmetry conditions.

### 5.4.4 FSI Simulations

With the groundwork of the individual solvers and mesh mapping presented, the next step is the analysis of the coupled simulations. Two flow conditions are considered, where the angle of attack is set to $0^o$ and $9^o$. The simulations are very similar in setup, with the only major difference being the use of symmetry conditions for computational efficiency. Both simulations were run using the loosely coupled, steady-state approach, where individual solver convergence was achieved before passing surface boundary condition data between solvers. The CFD solvers were initially run for 6,000 iterations to provide sufficient time for the flow to develop. Following these initial iteration counts, each subsequent FSI iteration ran for 2,000 iterations.

#### *5.4.4.1 $0^o$ Angle of Attack Simulation*

The simulation was run using one quarter of the geometry, as a result of leveraging symmetry planes. The initial cell count was 93,334, and at the completion of the simulation rose to 714,716 cells. A total of 10 FSI iterations were performed, resulting in a total computational time of approximately 31 hours on a hyperthreaded quad core CPU, equating to 8 OpenMP threads. A flowfield solution is shown in Fig. 93, where contours are displayed along with the computational mesh.

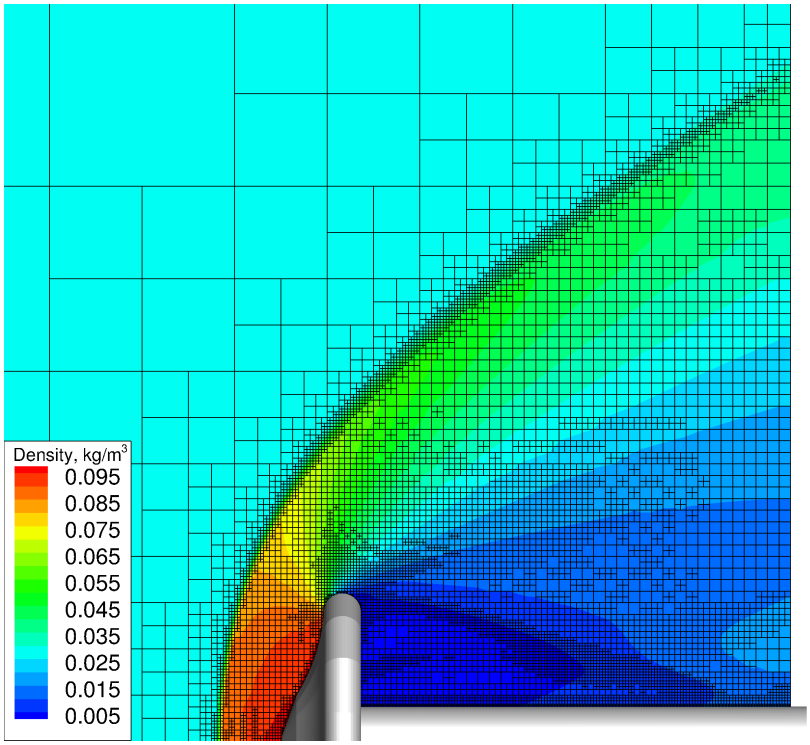



Figure 93: Flowfield slice along symmetry plane showing final density solution for the $0^o$ angle of attack.

The symmetry plane shapes are compared in Fig. 94, where the outlines are shown for the initial and final configurations, and are compared against the experimental data. The computed deflections under-predict compared to the data, which is in agreement with the computational analysis presented by Tanner [107]. The test article underwent a $0.94^o$ nose up rotation, which was not intended to be captured in the computational analysis, as it was constrained by the symmetry plane boundary conditions. Even without the use of symmetry planes, this rotation would not occur numerically, as the modeling of this problem results in a symmetric solution.

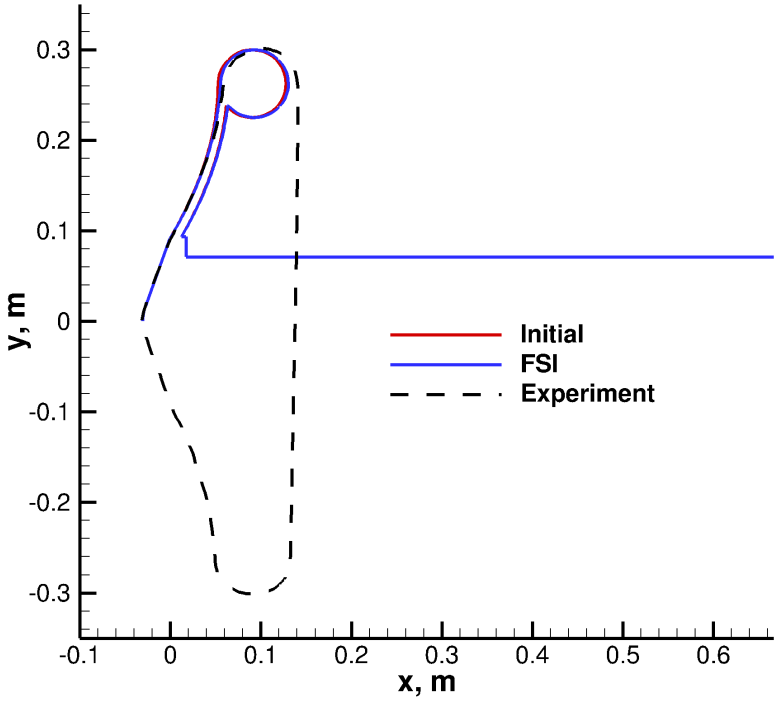



Figure 94: Comparison of symmetry plane deformation between initial geometry, FSI converged geometry, and experimental data.

The simulation convergence history is illustrated in Figs. 95, 96, and 97. The normalized RMS residual in Fig. 95 shows the clear impact of the initial geometry motion after 6,000 iterations, where the geometry is deformed for the first time. After this event, the displacements are sufficiently small when compared against the high frequency oscillations caused by mesh adaption. The axial force coefficient plot in Fig. 96 also exhibits the large initial variations in $C_A$ that quickly converge. The axial force coefficient converges after the first FSI iteration, and remains fairly constant for the remainder of the simulation. The torus displacement undergoes its maximum change in displacement during the first FSI iteration, and continues on a downward trend for the remainder of the simulation.

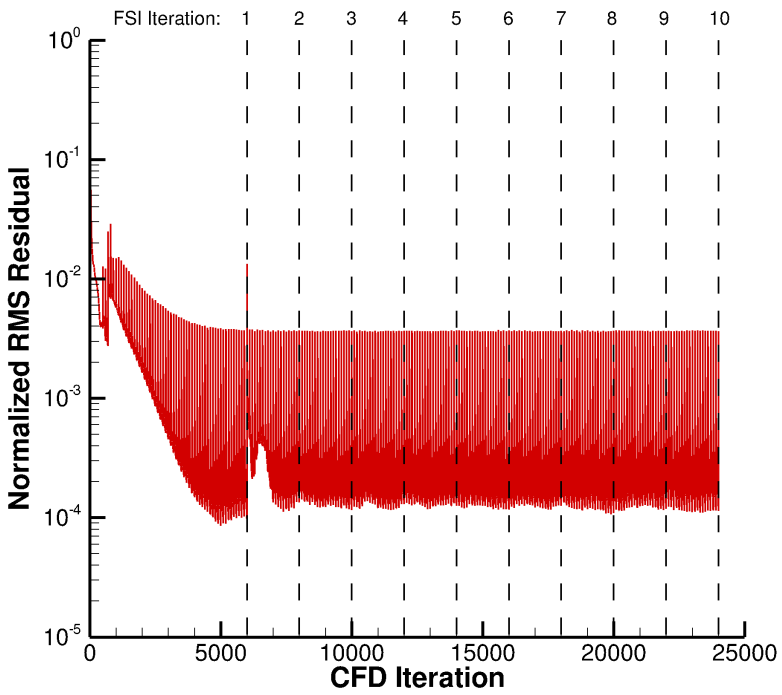



Figure 95: CFD normalized RMS residual history.

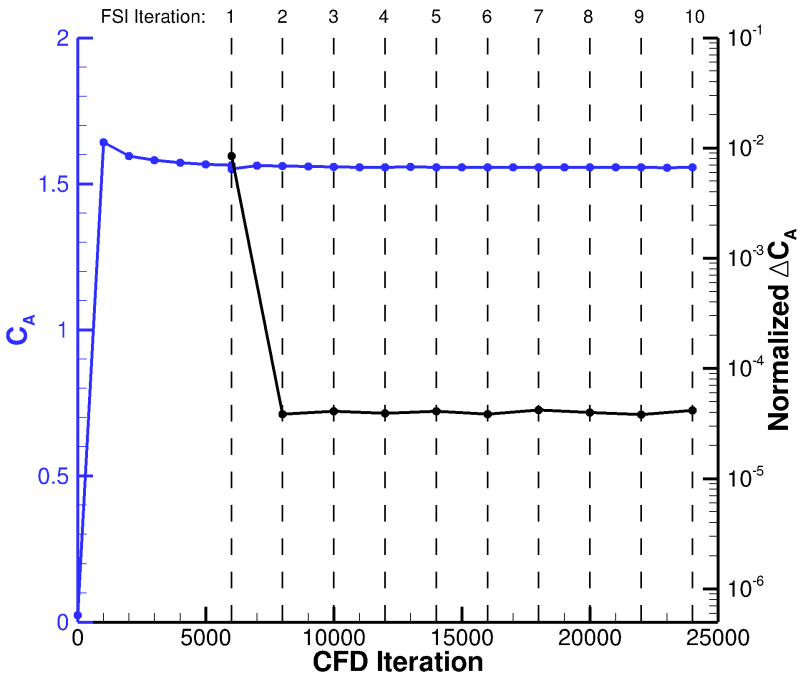



Figure 96: CFD axial load history computed over entire input geometry configuration.

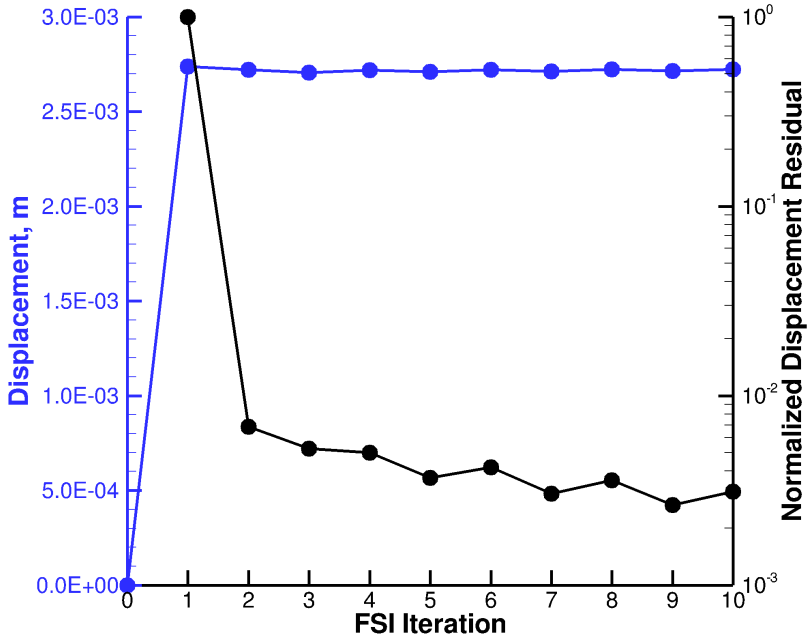



Figure 97: Axial displacement history of the torus.

#### *5.4.4.2 $9^o$ Angle of Attack Simulation*

The next simulation was run at an angle of attack of $9^o$, requiring half of the input geometry with a single symmetry plane. This required far more computational cells, with the initial count at 186,020, and the final cell count at 1,467,839. The simulation required 63 CPU hours to complete. As with the $0^o$ angle of attack simulation, this was also run for 10 FSI iterations.

Figure 98 shows the flowfield slice along the symmetry plane. The asymmetric flow pattern, which is caused by the angle of attack, results in an altered bow shock compared to the $0^o$ angle of attack case. With the bow shock closer to the tension cone on the lower portion, it is expected that the surface pressure will be larger, and lead to greater forces acting in this region. This will ultimately yield larger deflections here as well.

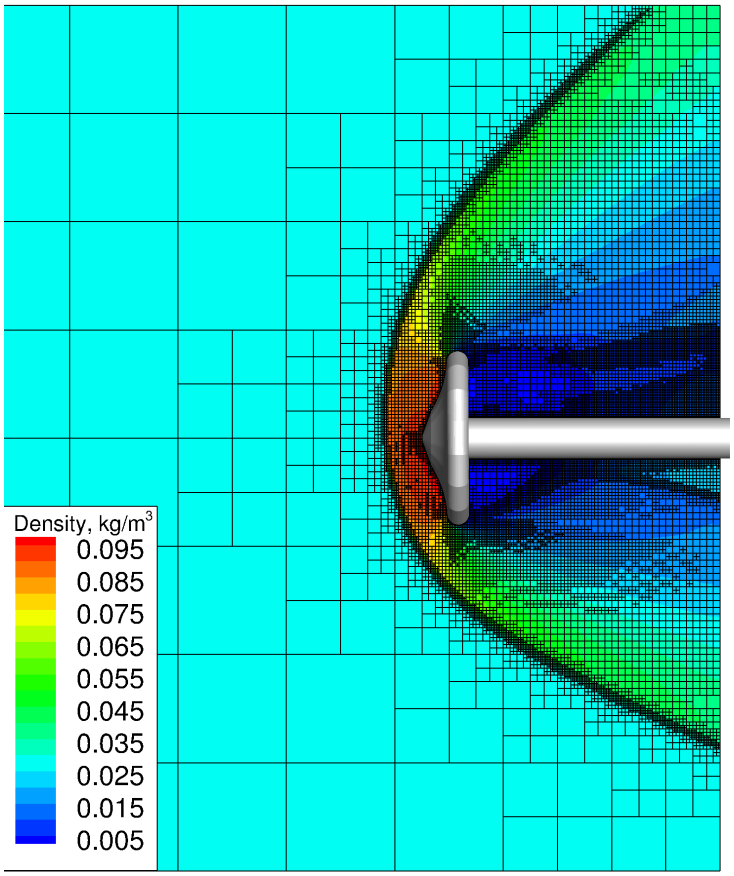



Figure 98: Flowfield slice along symmetry plane showing final density solution for the $9^o$ angle of attack.

A comparison between the initial input geometry and the final deformed geometry is shown in Fig. 99. The deflections are very small, and barely perceptible on the upper portion, while the lower portion shows clear deformations. Since the pressure distribution is no longer symmetric, the rigid torus rotates under this load, where the rotation is quantified as a measure of the angle the torus makes with the vertical plane. The calculated angle is approximately equal to $0.32^o$, compared to the measured angle of approximately $1^o$ for the test article. However, the angle produced in the present simulation closely matches that of the computational result presented in Ref. [107] of $0.3^o$. The comparison is shown in more detail in Fig. 100.

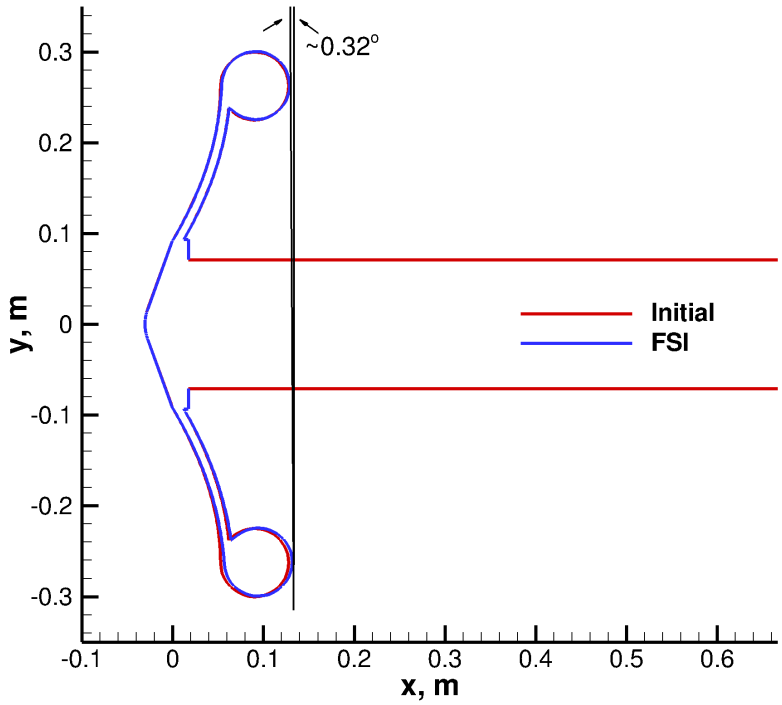
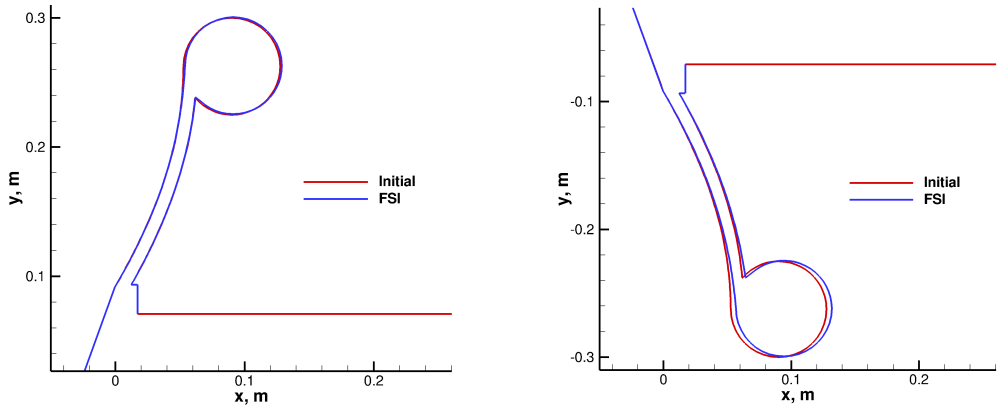



Figure 99: Comparison of symmetry plane deformation between initial geometry and FSI converged geometry for the $9^o$ angle of attack simulation.




(a) Upper portion.

(b) Lower portion.

Figure 100: Close-up views of the upper and lower portions of the tension cone for the $9^o$ angle of attack simulation.

The next three plots illustrate the convergence history of the simulation. Figures 101 and 102 show the normalized CFD residual and the axial force coefficient. These plots are

very similar to the ones shown for the $0^o$ angle of attack simulation, and provide the same conclusions, in terms of convergence. Figure 103 reports the displacement history of the torus, but does so by tracking two nodes on the torus. These nodes are located at the upper and lower end of the torus, centrally located on the symmetry plane. Both nodal displacement histories are shown in order to further illustrate the asymmetric deflection.

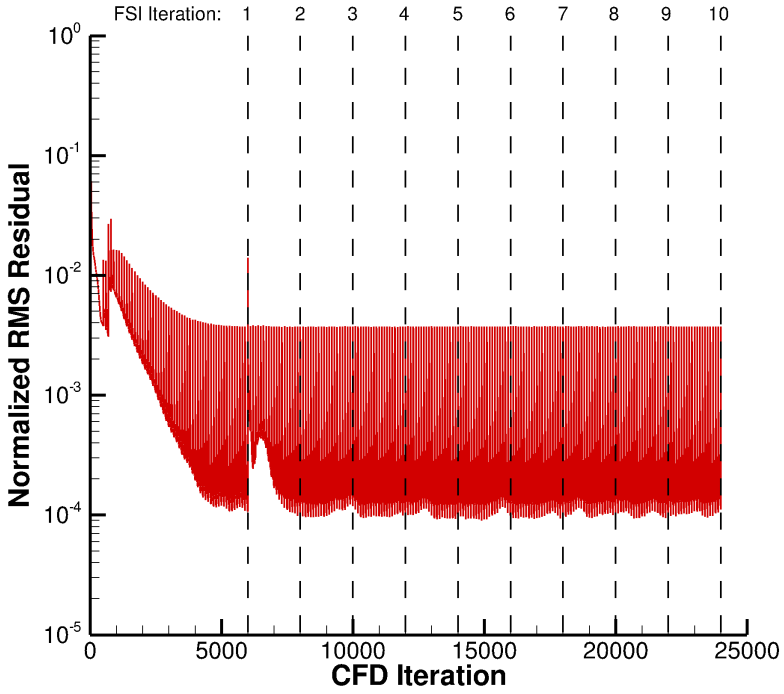



Figure 101: CFD normalized RMS residual history for the $9^o$ angle of attack simulation.

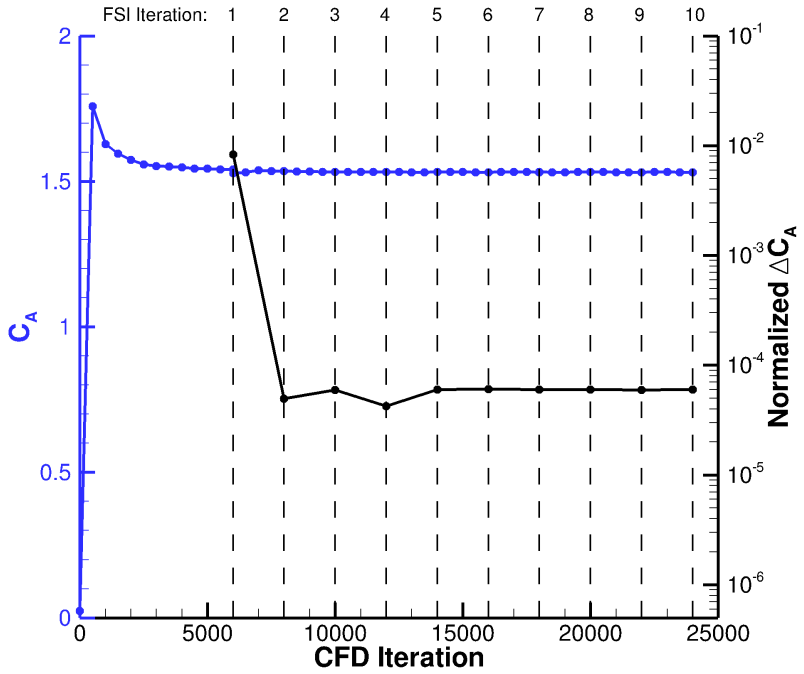



Figure 102: CFD axial load history computed over entire input geometry configuration for the $9^o$ angle of attack simulation.

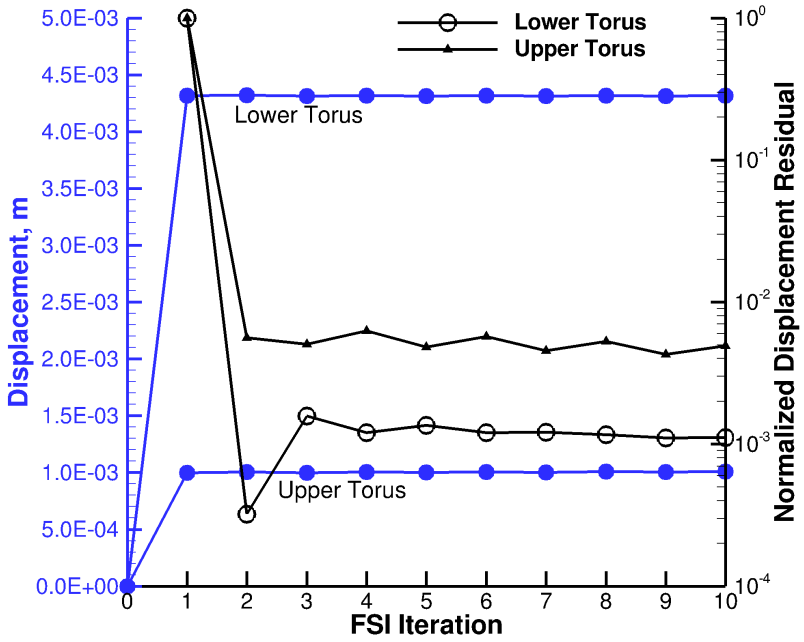



Figure 103: Axial displacement history of the torus for the $9^o$ angle of attack simulation.

## *5.5 Time Accurate Analysis of Semi-Rigid Tension Cone*

Time accurate, dynamic simulations of the semi-rigid tension cone are discussed next. At present, there exists little to no validation data with this application primarily because there have yet to be flight architectures that are designed to undergo strong fluid-structure coupling. A major exception is the use of parachutes, which can display significant dynamics caused by fluid-structure interactions. However, the objective of this work is to demonstrate the effectiveness of the methodology on future deceleration devices. It is with this justification, that simulations are presented demonstrating the fluid-structure interactions associated with notional conditions for the semi-rigid tension cone.

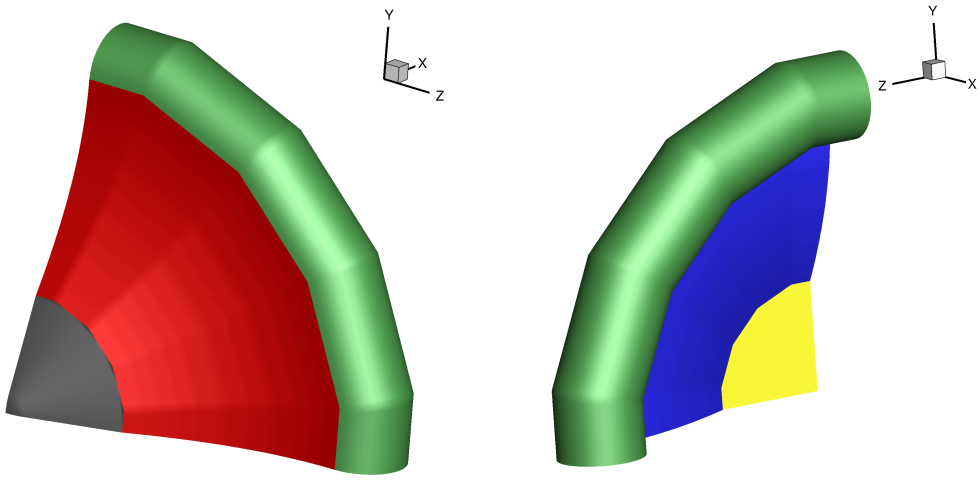
The objective of this section is to provide a set of simulations that demonstrate the effectiveness of the current tool to solve such problems. The fluid dynamics is modeled by considering flow at zero degree angle of attack, where symmetry planes can be utilized across the Y and Z planes. The inviscid, supersonic flow will not lead to any unsteady fluid dynamic phenomena. Unsteady motion is introduced by considering the dynamic behavior of the tension cone in response to the sudden aerodynamic loading experienced by the entry vehicle. The tension cone dynamically responds to the initial supersonic flow development produced during the CFD simulation. In the FSI simulations that will follow, the material thickness and freestream Mach number are varied in order to study the effects of these design parameters on the behavior of the system.

### 5.5.1 FEA Analysis

The FEA mesh, and nominal material properties remain unchanged from the steady-state analysis of the semi-rigid tension cone. The FEA analysis requires some adjustments to the input parameters in order to allow for dynamic motion, as opposed to the steady state equilibrium simulations that were previously run. The dynamic response has been computed using the dynamic implicit time stepping option. Specifically, the Newmark time integration scheme was used for all simulations. The $2^{nd}$ order accurate option was used, setting the scheme parameters to $\gamma = 0.5$ and $\beta = 0.25$.

### 5.5.2 CFD Analysis

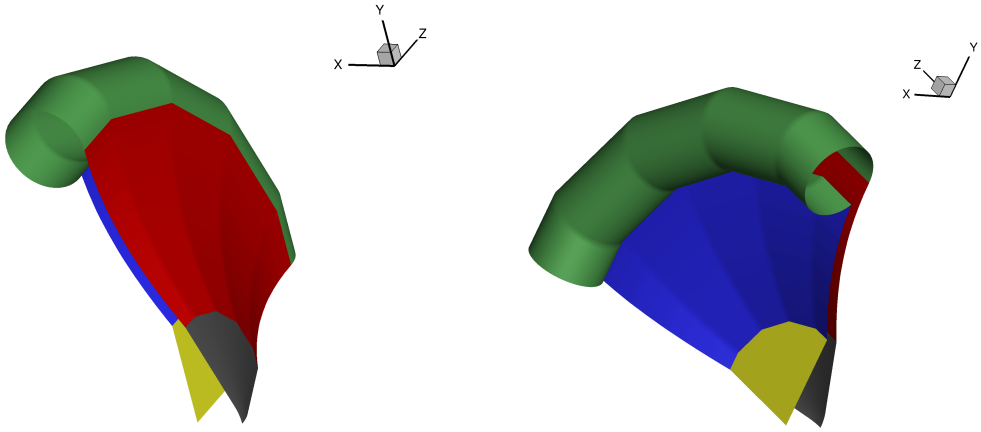
The CFD analysis of the tension cone is very similar to the setup used in the steady-state FSI simulations. However, an important change has been introduced to the configuration. The sting has been removed, and a backing plate has been added as a replacement part. This has been done primarily for two reasons. First, the removal of the sting significantly reduces the number of computational cells required along the surface, and ultimately leads to faster simulation times. Second, removing the sting leads to a configuration that is more representative of an actual flight scenario, rather than a wind tunnel setup. Several views of the CFD input geometry are shown in Figs. 104 and 105. The CFD simulations were advanced in time using the $3^{rd}$ order accurate TVD Runge-Kutta time stepping, using a constant time step. The inviscid fluxes were computed using the $1^{st}$ order accurate unstructured gradient approach for its improved computational efficiency. Solution adaption was performed every 10 iterations, and was based on velocity divergence. All other CFD parameters were set to the same values as for the steady-state simulations.




(a) Isometric view from the front.

(b) Isometric view from the rear.

Figure 104: CFD input geometry configuration consists of six components: 1. aeroshell (gray), 2. tension shell (red), 3. torus (green), 4. rear tension shell (blue), 5. backing plate (yellow).




(a) Isometric view from the front.

(b) Isometric view from the rear.

Figure 105: CFD input geometry as viewed from the opposite side of the symmetry plane. These views illustrate how the tension cone is constructed for the time accurate simulations.

### 5.5.3 FSI Analysis

The modification of two design parameters have been considered in the following analyses. The thickness, $h$, of the tension shell is considered by simulating 3 thicknesses: 1) the nominal material thickness that was previously used in the steady-state simulations, 2) half the thickness, and 3) twice the thickness. Payload mass is a critical design parameter on any space mission, and reducing the material thickness can help to minimize the mass. However, modifying the material thickness leads to changes in the dynamic response of the configuration, and the ability to predict this response is critical. Additionally, the freestream Mach number is also varied in order to gain insight into how this affects the response. In the simulations, the freestream Mach number is changed, while leaving the freestream pressure and temperature unchanged. This leads to a variation in freestream density, and has the effect of either increasing or decreasing the aerodynamic load on the vehicle.

Throughout the following discussions, the nominal simulation is referred to as the one for which the material thickness ($h_o$) and freestream Mach number is the same as was used during the steady-state analyses. This is considered the baseline simulation. The simulations where material thickness is varied are identified as $h = 0.5h_o$ and $h = 2h_o$, for a reduction in thickness and increase in thickness, respectively. The off-nominal freestream Mach numbers

were chosen to halve and double the freestream dynamic pressure respectively. These Mach numbers correspond to $M = 1.73$ and $M = 3.45$.

A snapshot of the flowfield for the nominal case is shown in Fig. 106, where the symmetry plane flowfield is colored by Mach number, and the surface of the tension cone by pressure coefficient. For these simulations, the rigid torus undergoes relatively small axial displacements, as a function of the tension shell motion. The majority of the structural response is constrained to the motion of the tension shell.

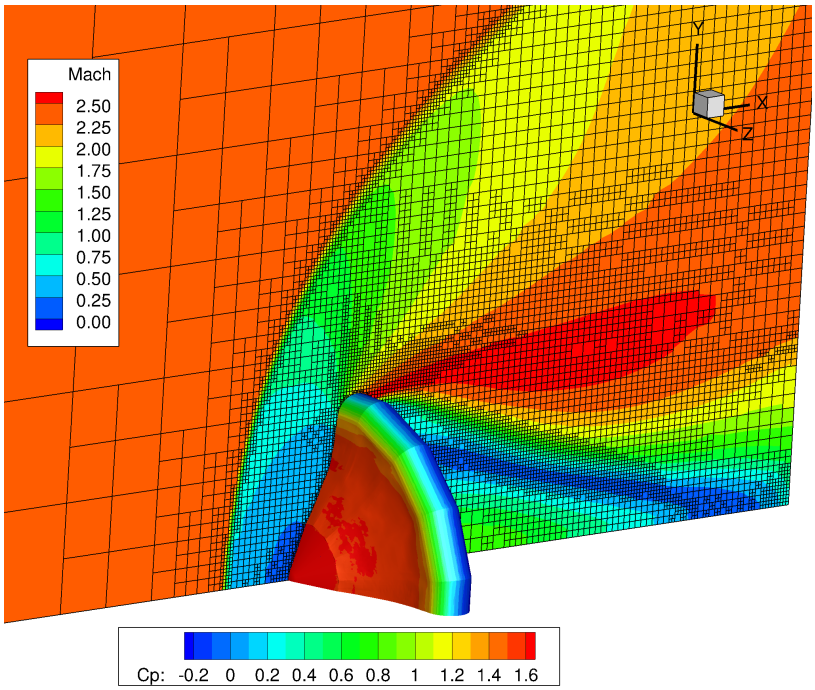
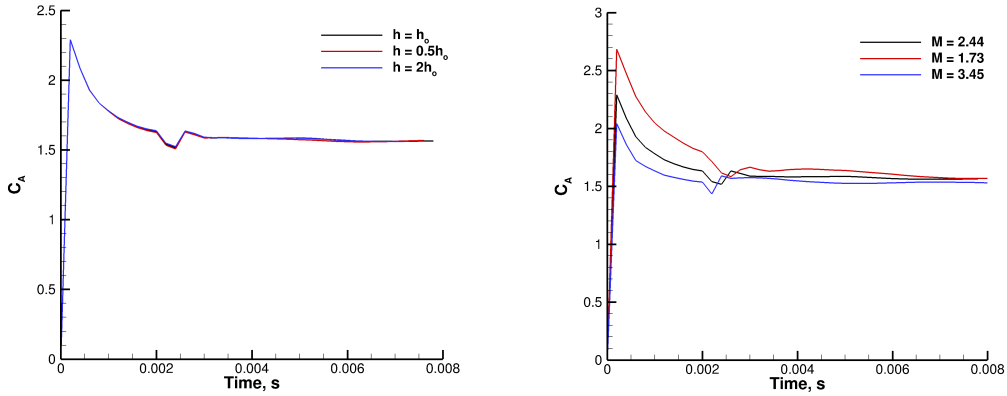



Figure 106: Flow solution of nominal dynamic FSI simulation. The flowfield is colored by Mach number and the surface by pressure coefficient.

The axial force coefficients for these simulations are shown in Fig. 107. The effects of material thickness are shown in Fig. 107a, where the initial axial load rises quickly as the initial flowfield develops, and bow shock is generated in front of the tension cone. As the flow develops, the axial force coefficient begins to fall, and undergoes some abrupt changes when solution adaption is initiated. This process significantly improves the accuracy of the bow shock shape and stand-off distance. From there, $C_A$ asymptotes towards a value of $C_A \approx 1.56$. All three plots nearly lie on top of each other, indicating that the chosen

variations in material thickness have little to no effect on the aerodynamic loads acting on the vehicle. This result is expected for two reasons. First, changing the material thickness results in local oscillations in displacement of varying amplitude. However, the overall mean motion of the tension cone remains largely unchanged. Second, the magnitude of the oscillations are still relatively small compared to the length scales of the entire tension cone, producing localized effects near the surface of the vehicle, but not large enough to effect the overall flow dynamics significantly. The effects of freestream Mach number are shown in Fig. 107b. The initial rise in $C_A$ for each simulation is now different, primarily due to the changes in initialization caused by different freestream Mach numbers. The axial force coefficients for each simulation are within a 3% window of each at the end of the simulations.



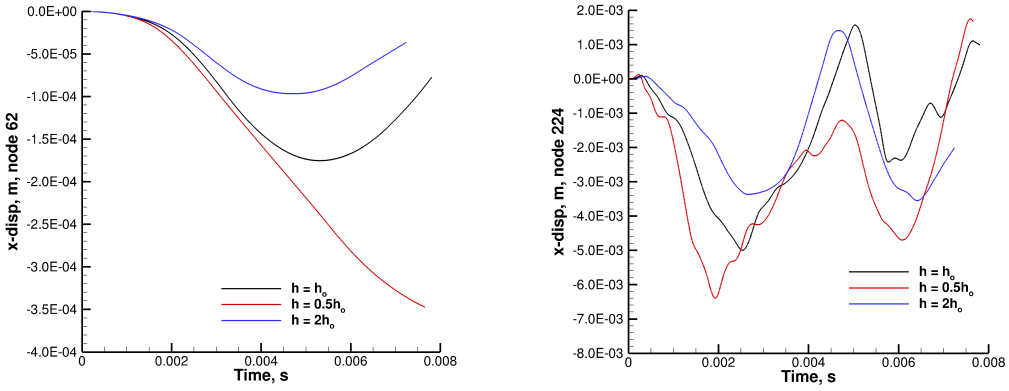

(a) (b)

Figure 107: Axial force coefficient of the entire configuration plotted against simulation time. The effects of material thickness are shown in (a), while the effects of freestream Mach number are shown in (b).

Next, a series of plots are shown to highlight the dynamic response of a select group of nodes and elements along the symmetry plane. These will serve to exemplify the typical response of any given location of the tension cone. These cannot fully characterize the dynamics, but are useful for identifying amplitudes and frequencies of motion. The displacement plots are time histories based on nodal values, while the stress plots are computed at over elements. The two nodes that are highlighted are nodes 62 and 224, which correspond to the tension shell-torus interface and the mid-location along the symmetry

plane, respectively. Elements 3126 and 2886 correspond to the same locations, and are used in plots where Von-Mises stress histories are shown.

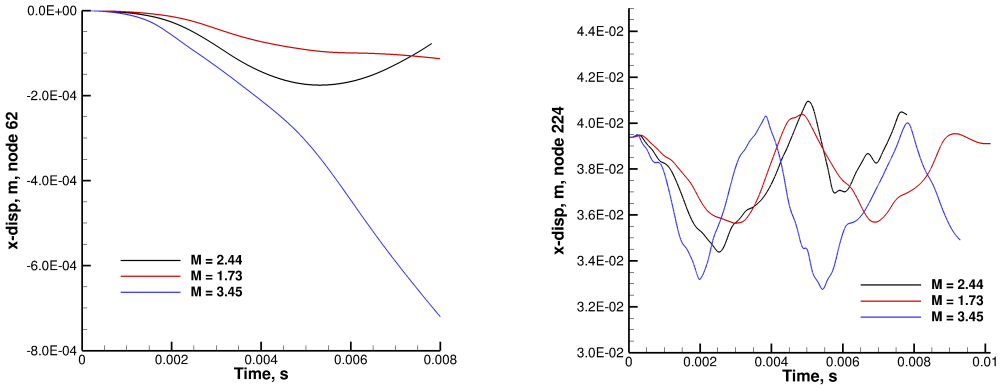
The first plots look at the axial displacement history of two nodes along the centerline as a function of material thickness. Figure 108a shows the displacement of the node that connects the tension shell to the torus, which effectively provides a means for tracking the displacement of the torus. As the material thickness decreases, the displacement magnitude and response frequency both increase. A similar trend is also seen in Fig. 108b.




(a) (b)

Figure 108: Comparison of nodal displacement in the axial direction as a function of material thickness. Node 62 is shown in (a) and node 224 in (b).

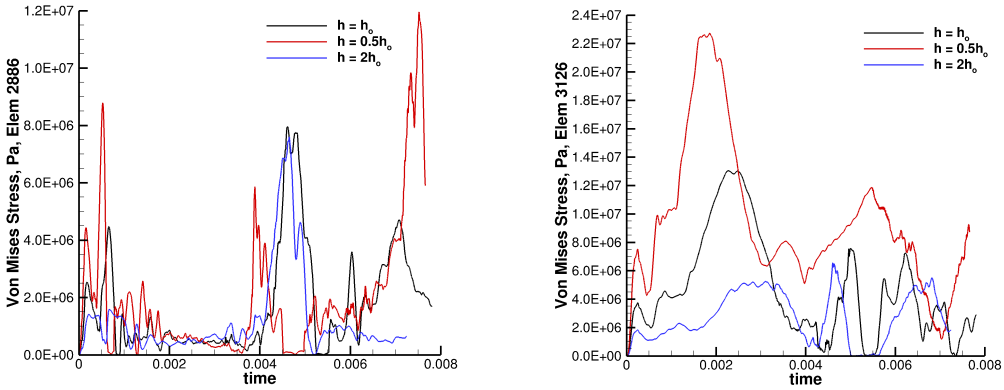
The effects of freestream Mach number on axial displacement are shown in Fig. 109. Increasing the freestream Mach number leads to larger displacements of the torus, due to the larger dynamic loading. Figure 109b demonstrates that as the freestream Mach number increases, so do the amplitudes and frequencies of the structural response.




(a) (b)

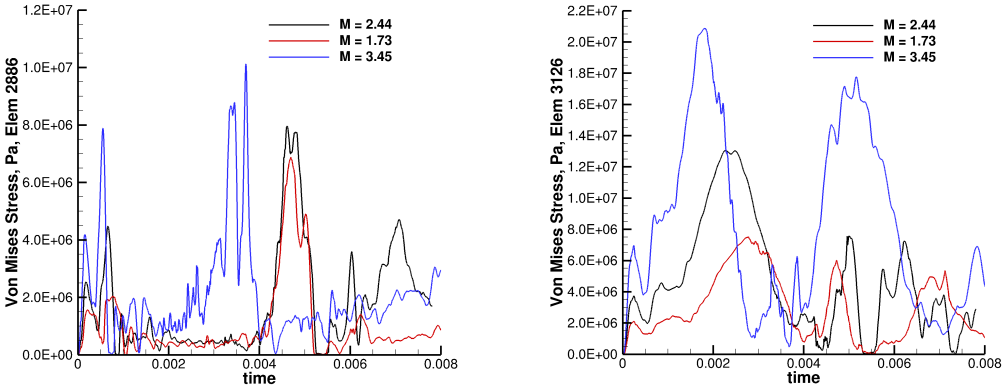
Figure 109: Comparison of nodal displacement in the axial direction as a function of freestream Mach number. Node 62 is shown in (a) and node 224 in (b).

Von Mises stress is shown in Fig. 110 as a function of material thickness. Reducing the material thickness results in the expected increase in stress. Additionally, the effects of freestream Mach number are shown in Fig. 111. Once again, the expected trends are observed where increases in Mach number, which lead to increases in aerodynamic loads, produce larger values of stress.




(a) (b)

Figure 110: Comparison of Von-Mises stress as a function of material thickness. Element 2886 is shown in (a) and element 3126 in (b).




(a) (b)

Figure 111: Comparison of Von-Mises stress as a function of freestream Mach number. Element 2886 is shown in (a) and element 3126 in (b).

# CHAPTER VI

# CONCLUSIONS AND FUTURE WORK

## *6.1 Overview*

The work presented in this document fulfills the objectives set forth in the introduction, which has advanced the state of the art in the computational modeling of fluid-structure interaction problems. The goal of this endeavor has been to develop and validate a time accurate aeroelastic methodology using a Cartesian immersed boundary method for the analysis of aerodynamic decelerators. This has been accomplished through the use of a partitioned coupling approach, where the fluid dynamics was solved using the in-house computational fluids dynamics software, NASCART-GT, and the commercially available structural dynamics tool, LS-DYNA. This research advances the state of the art in fluid-structure interaction simulations by addressing the following items.

1. **Characterized numerical accuracy of the non-stationary immersed boundary method applied in an unstructured Cartesian grid flow solver.** The spatial accuracy has been explored by studying the surface and off-surface physics as a function of the surface motion, and in particular its velocity. The spatial order of accuracy has been reported, and comparisons in error have been made between stationary and non-stationary surfaces. A 1-D study was presented that characterized the numerical error as a function of surface velocity. Such analyses have not been found in the literature, and this work highlights these results in order to gain insight into these effects, which are crucial in determining appropriate mesh resolution for moving body and FSI simulations.

2. **Developed an FSI methodology that fills a mid-fidelity gap in the computational analysis of atmospheric entry configurations.** A mid-fidelity, highly automated, computational framework for analyzing fluid-structure interactions has been developed. Other FSI software packages that are presented in the literature often require hands-on mesh generation, and only allow small deflections. The approach described in

this body of work is capable of automatically generating the CFD mesh, adapting the mesh where appropriate, and allows large structural displacements. The methodology has been enhanced through the development of an efficient computational framework. The computational fluid dynamics software has been re-written, with the goal of improving computational efficiency, as well as code modularity and ease of future development. The combination of the methodology and software design, lays the groundwork for improved simulation capability, as well as the continued development of numerical features.

3. **Demonstrated time accurate fluid-structure interaction simulations of the dynamic motion of a tension cone.** Simulations were presented, which demonstrate the time accurate coupling methodology applied to a relevant atmospheric entry configuration. This research extends prior FSI work, which studied the steady-state properties of the tension cone IAD in supersonic flow. Despite the lack of experimental data for this particular problem, the present work has explored the effects of material properties and freestream Mach number on the behavior of the tension cone. Confidence in the numerical accuracy of these results has been achieved through the analysis of one-dimensional and two-dimensional FSI problems, for which validation data is present.

## *6.2 Results & Conclusions*

The first section of results presented in this thesis explored the numerical accuracy of stationary and prescribed motion CFD simulations. Supersonic flow over a rigid tension cone demonstrated the inviscid capability of the solver to accurately predict the shock structure and the surface pressure distribution. A series of non-stationary, prescribed motion simulations studied the effects of moving bodies. These simulations addressed numerical accuracy concerns, and how the solution-based mesh adaption played an important role in solution accuracy. The first set of simulations examined a 1-D translating piston, providing a means for studying the effects of the non-stationary boundary conditions. Spatial order of accuracy was studied by looking at the effects of mesh resolution. It was found that in order to maintain similar accuracy to stationary input geometries, a finer mesh resolution was required to capture the propagation of shock waves. As the speed of the translating

piston increased, the accuracy in the compression region was reduced, while the accuracy on the expansion side was increased. Supersonic flow past a translating diamond airfoil and a cylinder provided an additional means of characterizing the effects of moving bodies. The purpose of these simulations was to compare equivalent flowfields, where the frame of reference was changed. By studying flow on uniform and non-uniform meshes, it was found that the robustness of solution-based grid adaption was a critical component in accurately predicting the flowfield. Finally, the pitching motion of a NACA 0012 airfoil was studied in transonic flow. Comparisons against experimental data demonstrated the accuracy in computing the aerodynamic loads history.

The next step towards FSI problems was examined by studying the rigid body dynamics of the MSL capsule. This free flight motion is a specialized example of FSI, where there are no elastic modes, but the configuration is allowed to undergo rigid body motions. A 2-D surrogate for the MSL capsule was studied, and results were compared against reference numerical results. Static simulations provided validation of the CFD solver to accurately compute static loads. A free flight analysis was performed, where similar meshes were used between the present simulations and the reference data. Agreement in pitching frequency between the present results and reference data was established. It was then shown with further mesh refinement, that a grid converged solution produced a higher frequency response, with negligible pitch damping over the course of the simulations.

Time accurate FSI simulations were studied by investigating several physical problems. The first set of analyses studied the interaction between a compressible fluid in a chamber and a piston attached to a spring. A simplified 0-D fluid model provided a means for validating the numerical accuracy for several coupling strategies up to $2^{nd}$ order accuracy. The 1-D problem was then studied, where the CFD solver was used to compute the loads applied to the piston. Several coupling schemes were studied, and the numerical error demonstrated at most $1^{st}$ order accuracy. Despite the fact that the structural dynamics and coupling schemes were up to $2^{nd}$ order accurate in time, the CFD simulations only demonstrated $1^{st}$ order accuracy, thus reducing the global order of accuracy.

The complexity was then increased by studying 2-D flows. Incompressible, viscous flow

around an elastic cylinder generates vortex shedding, and thus a time varying oscillatory loads and displacement history. The FSI simulation produced transverse displacement amplitudes of the correct magnitude. However, the mean axial displacement was over-predicted due to the insufficient mesh resolution caused by the stringent requirements on time step and total simulation time. Supersonic flow was then studied where a shock wave interacted with a flexible panel. Two panel lengths were studied, where the shorter panel simulation produced displacement histories in agreement with experimental data and reference numerical results.

Finally, several FSI simulations were presented for a tension cone configuration. Steady-state analyses at two angles of attack highlighted the ability of the methodology to accurately predict the deformed configuration shape along the symmetry plane, when compared against experimental and numerical results. Time accurate simulations of this entry vehicle served to address the final contribution of this work. The dynamic response for these simulations was characterized by studying the displacement and stress histories, as well as the aerodynamic loads over time. The effects of Mach number and material thickness were explored, demonstrating that the present methodology can serve as a predictive computational tool for the analysis of inflatable decelerators.

## *6.3 Suggestions for Future Work*

The present work explored many facets to a large, challenging problem in the field of computational simulation and modeling. Thus, not all concerns could be addressed, resulting in the following discussion regarding suggestions for future work.

Validation data for time accurate FSI problems is critical, as continued validation of the present computational methods are pursued. These include experimental data, as well as flight test data. As this data becomes available, flight relevant flow and material properties can be selected and studied. In addition, the predictive nature of these computational analyses can facilitate design decisions for experimental configurations.

Upon initial investigation of structural solvers, LS-DYNA was deemed sufficient for the

simulation of FSI problems within the current framework. However, during the development of the time accurate algorithms, there were several roadblocks that were encountered. There is currently no user interface within LS-DYNA to easily facilitate the development of software coupling. Interface code was added within a user defined loading library to force this process to work. A suggestion for future work is two-fold. First, continued development of this tool with LS-DYNA would significantly benefit from collaboration with the LS-DYNA developers. The addition of coupling interfaces to LS-DYNA would potentially allow for higher order coupling approaches to be implemented. Higher order coupling methods require more precise control over the advancement in time of the simulation, or in the case of a tightly coupled, sub-iteration approach, the simulation requires multiple sweeps at a given time step. Additionally, other software packages could be considered where more advanced coupling interfaces exist. For steady-state coupling, the use of LS-DYNA has demonstrated its effectiveness. However, for time accurate coupling, these challenges must be considered by future developers.

The order of accuracy of the CFD simulations within NASCART-GT must be explored in greater depth, in terms of both space and time. Surface boundary conditions are currently enforced using a $1^{st}$ order accurate approach, which limits the global order of accuracy. Within this study, conservation of mass, momentum, and energy should be considered, as these properties are not currently enforced at the surface. Improvements to the spatial order of accuracy may lead to the ability to formally demonstrate the appropriate temporal accuracy that is expected for the implemented schemes. With this enhanced foundation, the numerical accuracy of higher order FSI problems can be studied in much greater detail.

It has been shown in the present work that the quality of non-uniform meshes plays a critical role in the accuracy of the solution, especially for non-stationary geometries. Further investigations into the numerical accuracy in these regions of the flow are critical. Additionally, it has been shown that computational time becomes a constraint when running time accurate simulations using an explicit time scheme. The development of a computationally efficient implicit time scheme within NASCART-GT will significantly enhance the presently developed capabilities.

With regards to the CFD meshing capabilities in NASCART-GT, one of the challenges with FSI problems is working with the different mesh topologies between solvers. The ability to represent infinitely thin membrane surfaces within NASCART-GT would simplify the FSI mesh mapping process, since the structural mesh could be used as the input geometry to the CFD simulation. Though desirable, this technique brings many challenges associated with the representation of the surface, as well as the application of the boundary conditions within the CFD code.

# APPENDIX A

## ALE FORMULATION OF AUSMPW+

The AUSMPW+ flux function is given by

$$F_{\frac{1}{2}} = \bar{M}_L^+ c_{\frac{1}{2}} \mathbf{\Phi}_L + \bar{M}_R^- c_{\frac{1}{2}} \mathbf{\Phi}_R + \left( P_L^+ |_\alpha \left[ \mathbf{P}_L \right] + P_R^- |_\alpha \left[ \mathbf{P}_L \right] \right) \tag{51}$$

The vector, $\Phi$, is given by

$$\mathbf{\Phi} = \left[ \rho, \rho u, \rho v, \rho w, \rho h \right]^T \tag{52}$$

$$\mathbf{P} = \left[ 0, p, 0, 0, 0 \right]^T \tag{53}$$

For the case when $m_{1/2} \geq 0$

$$\begin{aligned} \bar{M}_L^+ &= M_L^+ + M_R^- \left[ \left( 1 - \omega_p \right) \left( 1 + f_R \right) - f_L \right] \\ \bar{M}_R^- &= M_R^- \omega_p \left( 1 + f_R \right) \end{aligned} \tag{54}$$

and when $m_{1/2} \leq 0$

$$\begin{aligned} \bar{M}_L^+ &= M_L^+ \omega_p \left( 1 + f_L \right) \\ \bar{M}_R^- &= M_R^- + M_L^+ \left[ \left( 1 - \omega_p \right) \left( 1 + f_L \right) - f_R \right] \end{aligned} \tag{55}$$

The cell face Mach number, $m_{1/2}$, is defined by:

$$m_{1/2} = M_L^+ + M_R^- \tag{56}$$

The pressure weighting term is given by

$$\omega_p\left(p_L, p_R\right) = 1 - \min\left(\frac{p_L}{p_R}, \frac{p_R}{p_L}\right)^3 \tag{57}$$

and

$$f_{L,R} = \begin{cases} \left(\frac{p_{L,R}}{p_s} - 1\right) \min\left[1, \frac{\min\left(p_{1,L}, p_{1,R}, p_{2,L}, p_{2,R}\right)}{\min\left(p_L, p_R\right)}\right]^2, & p_s \neq 0 \\ 0, & p_s = 0 \end{cases} \tag{58}$$

where

$$p_s = P_L^+ p_L + P_R^- p_R \tag{59}$$

The Mach number and the pressure splitting functions are then provided by

$$M^{\pm} = \begin{cases} \pm\frac{1}{4}\left(M \pm 1\right)^2, & |M| \leq 1 \\ \frac{1}{2}\left(M \pm |M|\right), & |M| > 1 \end{cases} \tag{60}$$

and

$$P^{\pm}|_{\alpha} = \begin{cases} \pm\frac{1}{4}\left(M \pm 1\right)^2\left(2 \mp M\right) \pm \alpha M\left(M^2 - 1\right)^2, & |M| \leq 1 \\ \frac{1}{2}\left(1 \pm \text{sign}\left(M\right)\right), & |M| > 1 \end{cases} \tag{61}$$

The Mach number from the right and left of the interface is

$$M_{L,R} = \frac{U_{L,R}}{c_{\frac{1}{2}}} \tag{62}$$

and the cell interface speed of sound is:

$$c_{\frac{1}{2}} = \begin{cases} c_s'^2 / \max\left(|u_L|, c_s'\right), & \frac{1}{2}\left(u_L + u_R\right) > 0 \\ c_s'^2 / \max\left(|u_R|, c_s'\right), & \frac{1}{2}\left(u_L + u_R\right) < 0 \end{cases} \tag{63}$$

where

$$c_s' = \sqrt{2 h_{\text{norm}}\left(\gamma - 1\right)} \tag{64}$$

Upon consideration of rigid body mesh motion with velocity

$$\dot{\mathbf{x}} = (\dot{x}, \dot{y}, \dot{z}) \tag{65}$$

the equations are modified by a shift in convective terms. The speed condition in Eq. [63] must be adjusted such that Eq. [63] becomes

$$c_{\frac{1}{2}} = \begin{cases} c_s'^2 / \max\left(|u_L|, c_s'\right), & \frac{1}{2}\left(u_L + u_R - \dot{x}_{normal}\right) > 0 \\ c_s'^2 / \max\left(|u_R|, c_s'\right), & \frac{1}{2}\left(u_L + u_R - \dot{x}_{normal}\right) < 0 \end{cases} \tag{66}$$

Additionally, the definition of the left and right Mach number in Eq. [62] becomes

$$M_{L,R} = \frac{U_{L,R} - \dot{x}_{normal}}{c_{\frac{1}{2}}} \tag{67}$$

Finally, the energy equation requires a correction term to account for the fact that grid motion cannot contribute to pressure work. Since the modified Mach number is multiplied by $\Phi$ in Eq. [51], the enthalpy is convected at the modified Mach number. Only the internal energy component of the enthalpy can be affected by grid motion. Thus, the product of pressure and grid velocity must be added back in. This is taken in account in the pressure vector in Eq. [51] with the following modification:

$$\mathbf{P} = [0, p, 0, 0, p\dot{x}_{normal}]^T \tag{68}$$

# APPENDIX B

## 6-DOF MODEL

The following mathematical formulation of the 6-DoF model strongly parallels the description presented by Murman et al. [84]. The quaternion representing the orientation is given by:

$$\mathbf{p} = [e_0, e_1, e_2, e_3] \tag{69}$$

The transformation matrix to convert from the body reference frame to the global inertial frame is given in terms of the quaternion components:

$$\mathbf{A} = 2\begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1e_2 - e_0e_3 & e_1e_3 + e_0e_2 \\ e_1e_2 + e_0e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2e_3 - e_0e_1 \\ e_1e_3 - e_0e_2 & e_2e_3 + e_0e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix} \tag{70}$$

The inertia matrix must be provided as input, and is denoted by $\mathbf{I}^b$.

The applied moment ($\mathbf{M}$) to the system is computed in the global frame directly from the flowfield solution. The moment must be converted to the body frame:

$$\mathbf{M}^b = A^{-1}\mathbf{M} \tag{71}$$

The angular accelerations can be computed using Euler's equations in the body frame and solving for $\dot{\omega}^b$.

$$\begin{aligned} M_1^b &= I_1^b\dot{\omega}_1^b - (I_2^b - I_3^b)\omega_2^b\omega_3^b \\ M_2^b &= I_2^b\dot{\omega}_2^b - (I_3^b - I_1^b)\omega_3^b\omega_1^b \\ M_3^b &= I_3^b\dot{\omega}_3^b - (I_1^b - I_2^b)\omega_1^b\omega_2^b \end{aligned} \tag{72}$$

The angular accelerations are then integrated in time using a $2^{nd}$ order mid-point scheme, providing the angular velocities in the body frame.

The angular velocities can be converted to a quaternion rate through

$$\dot{p} = \frac{1}{2}\mathbf{L}^T \omega^b \tag{73}$$

The angular velocities are then used to update the actual orientation of the configuration over a given time step.

# APPENDIX C

# HARMONIC OSCILLATOR IMPLEMENTATION

A separate module for solving the mass-spring system has been implemented within NASCART-GT. The governing equation for a simple mass-spring-damper system is given by

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = f(u,t) \tag{74}$$

where $m$ is the mass, $c$ is the damping coefficient, $k$ is the spring stiffness, and $f$ is the forcing function. The displacement is given by $u$, and derivatives are with respect to time. The system is integrated numerically using the Newmark integration method, which is detailed in Appendix D. The implementation of this method is validated by examining free and forced oscillator motion, both assuming zero damping.

## *C.1 Free Mass Spring System*

Assuming a damping coefficient of zero and a forcing function equal to zero, the governing equation reduces to

$$m\ddot{u}(t) + ku(t) = 0 \tag{75}$$

where the general solution is given by

$$u(t) = A_1 cos(\omega_n t) + A_2 sin(\omega_n t) \tag{76}$$

where $\omega_n$ is the natural frequency, defined by $\omega_n = \sqrt{\frac{k}{m}}$. The constants, $A_1$ and $A_2$ are determined based on the initial conditions, $u(0) = u_o$ and $\dot{u}(0) = \dot{u}_o$, resulting in

$$u(t) = u_o cos(\omega_n t) + \frac{\dot{u}_o}{\omega_n} sin(\omega_n t) \tag{77}$$

The problem parameters and initial conditions were chosen to provide a physically similar character to the 1-D piston problem studied. The initial displacement is set to zero,

while the initial velocity is set to 20 $m/s$. The mass and spring stiffness are set to 0.8 $kg$ and 8,000 $N/m$ respectively.

The displacement history is shown in Fig. 112. Three time steps were studied, all resulting in nearly indistinguishable displacement histories, which is why only a single simulation is shown in the figure.

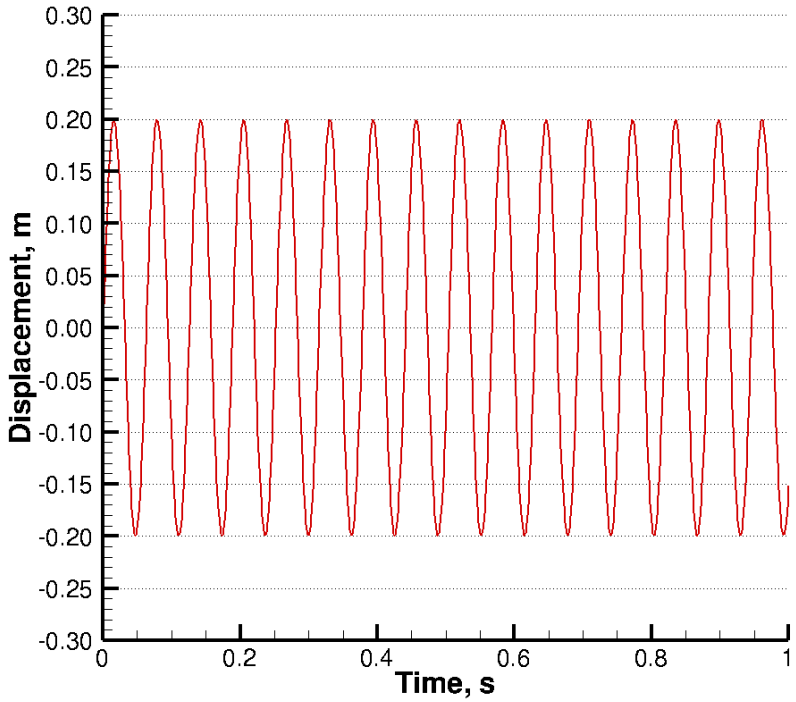



Figure 112: The displacement history is shown as a result of the numerical integration of the undamped, free response, mass-spring system.

The $L_1$ error is shown in Fig. 113. The difference in solution error is now clearly presented as a function of time step. The maximum error within each oscillation period is plotted as symbols with lines connecting the reported data. The error clearly diminishes along with the time step. Additionally, the error tends towards an asymptotic value as simulation time increases.

The order of accuracy is graphically represented in Fig. 114, where the error is plotted against the time step. The slope of the error on the log plot indicates the order of accuracy. For comparison, three separate instances in time have been presented, each with a slope just under a value of 2 (indicative of $2^{nd}$ order accuracy).

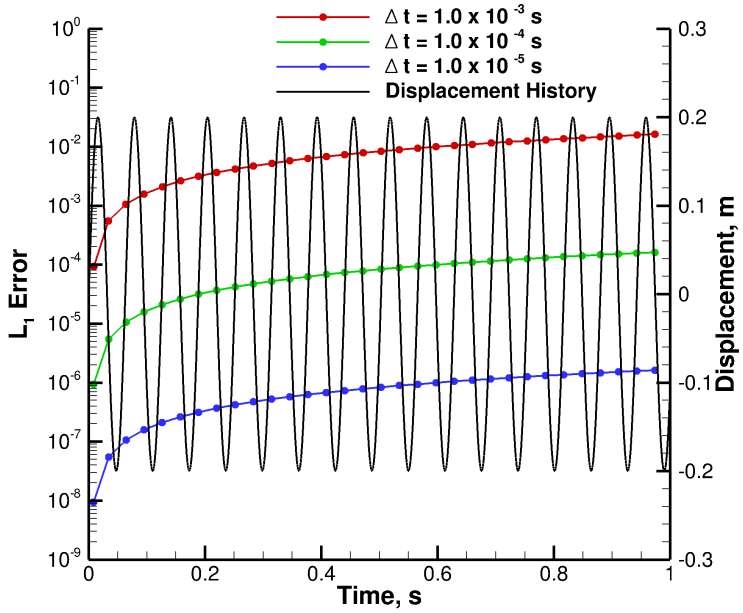



Figure 113: Time history of $L_1$ error. The displacement history is added to provide a reference. The $L_1$ error is shown for three different time steps, with the maximum error within each period of oscillation included. As the simulation progresses in time, the maximum displacement maintains a high level of accuracy compared to the points in time when the displacement is near zero, and the mass is moving at maximal velocity.

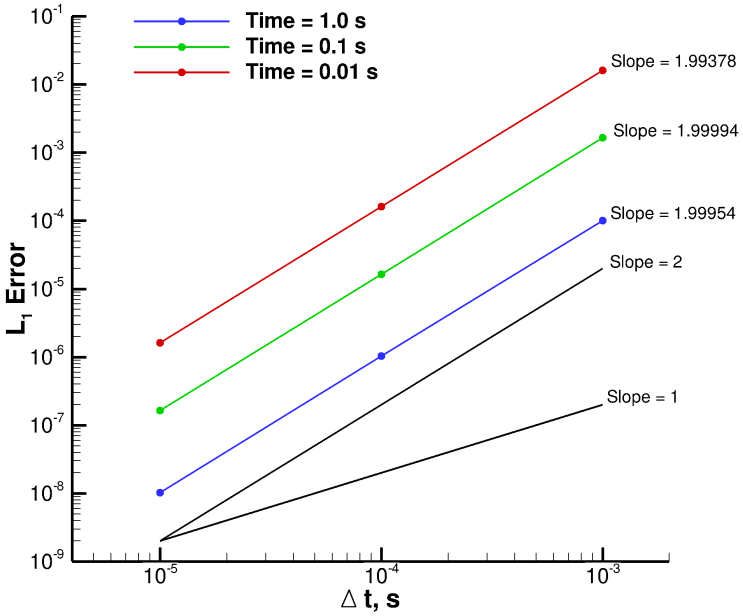



Figure 114: Time step convergence shown for three different time steps, processed at three separate instances in time.

## *C.2 Forced Mass Spring System*

Now, if the forcing function is permitted to be nonzero, while still maintaining zero damping, the governing equation becomes

$$m\ddot{u}\left(t\right)+ku\left(t\right)=f\left(u,t\right) \tag{78}$$

A sinusoidal forcing function is applied to the system, as this is similar to the expected forcing function in the 1-D FSI problem. The applied forcing function is chosen to be

$$f\left(u,t\right)=P_{o}cos\left(\Omega t\right) \tag{79}$$

where $P_o$ is the amplitude of the forcing function, $\Omega$ is the frequency of the driving force, and the general solution is given by

$$u\left(t\right)=A_{1}cos\left(\omega_{n}t\right)+A_{2}sin\left(\omega_{n}t\right)+\frac{P_{o}/k}{1-r^{2}}cos\left(\Omega t\right) \tag{80}$$

where $r=\frac{\Omega}{\omega_n}$. The constants, $A_1$ and $A_2$ are determined based on the initial conditions, $u(0)=u_o$ and $\dot{u}(0)=\dot{u}_o$, resulting in

$$u\left(t\right)=\left(u_{o}-\frac{P_{o}/k}{1-r^{2}}\right)cos\left(\omega_{n}t\right)+\frac{\dot{u}_{o}}{\omega_{n}}sin\left(\omega_{n}t\right)+\frac{P_{o}/k}{1-r^{2}}cos\left(\Omega t\right) \tag{81}$$

In order to specify the forcing function such that it represents the driving fluid motion of the corresponding FSI problem, the frequency and amplitude must be approximately matched. This requires a driving frequency of approximately $1,030rad/s$. Since the fluid pressure in the FSI problem is expected to vary up to $13,000Pa$ from the equilibrium condition of $101,325Pa$, a forcing amplitude of $13,000N$ is used (applied loading area equals $1m^2$). Thus, $P_o=13,000Pa$.

The $L_1$ error is shown in Fig. 116. The difference in solution error is now clearly presented as a function of time step. The maximum error within each oscillation period is plotted as symbols with lines connecting the reported data. The error clearly diminishes along with the time step. Additionally, the error tends towards an asymptotic value as simulation time increases.

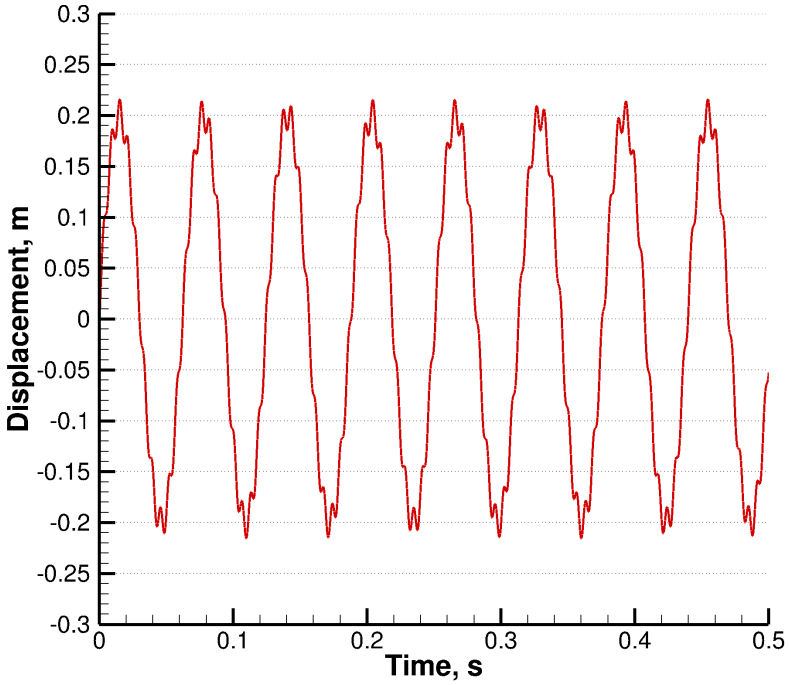



Figure 115: The displacement history is shown as a result of the numerical integration of the forced mass-spring system.

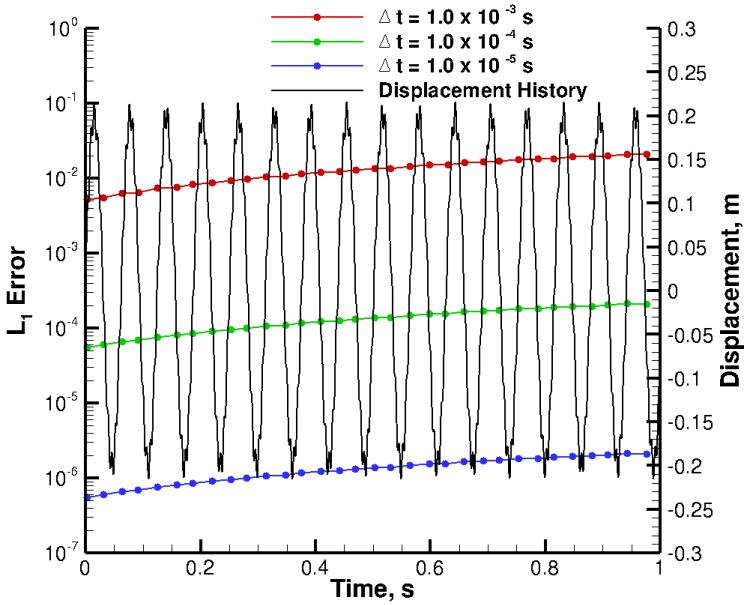



Figure 116: Time history of $L_1$ error. The displacement history is added to provide a reference. The $L_1$ error is shown for three different time steps, with the maximum error within each period of oscillation included.

The order of accuracy is graphically represented in Fig. 117, where the error is plotted against the time step. The slope of the error on the log plot indicates the order of accuracy. For comparison, three separate instances in time have been presented, each with a slope just under a value of 2 (indicative of $2^{nd}$ order accuracy).

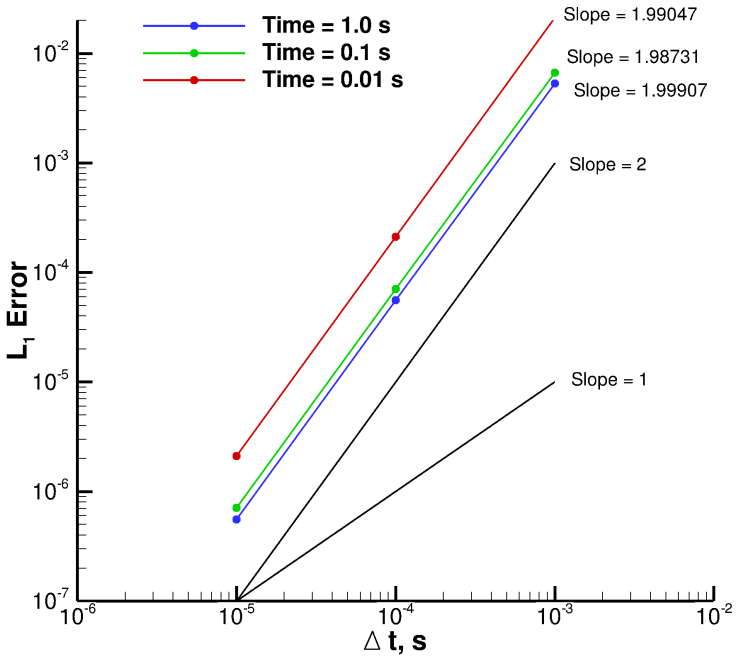



Figure 117: Time step convergence shown for three different time steps, processed at three separate instances in time.

# APPENDIX D

## NEWMARK INTEGRATION SCHEME

The Newmark method is a common method used for numerically solving differential equations, particularly with the dynamic response of structures. The method provides update equations for the state and its time derivative.

$$u^{n+1} = u^n + \Delta t \dot{u}^n + \Delta t^2 \left( \frac{1}{2} - \beta \right) \ddot{u}^n + \beta \Delta t^2 \ddot{u}^{n+1} \tag{82}$$

$$\dot{u}^{n+1} = \dot{u}^n + (1 - \gamma) \Delta t \ddot{u}^n + \gamma \Delta t \ddot{u}^{n+1} \tag{83}$$

The parameters, $\beta$ and $\gamma$, are set based on the desired numerical properties of the scheme. The constant acceleration method is used, where $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$.

Solving these equations requires a means for evaluating the acceleration at time $n + 1$. This is determined using the governing equations of interest, which in this case is that of the damped harmonic oscillator:

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = f(u, t) \tag{84}$$

Equations 82 and 83 are inserted into Eq. 84, resulting in

$$\begin{aligned} \left[ m + \gamma c \Delta t + \beta k \Delta t^2 \right] \ddot{u}^{n+1} = f^{n+1} - c \left[ \dot{u}^n + (1 - \gamma) \Delta t \ddot{u}^n \right] - \\ k \left[ u^n + \Delta t \dot{u}^n + \Delta t^2 \left( \frac{1}{2} - \beta \right) \ddot{u}^n \right] \end{aligned} \tag{85}$$

The equation is then solved for $\ddot{u}^{n+1}$ and the solution to the system is complete.

# APPENDIX E

## ENTROPY WAVE ADVECTION

A discussion on the accuracy of the numerical schemes is presented by considering the advection of an entropy wave. The purpose of this section is to demonstrate the implemented order of accuracy compared to the designed order of accuracy of the flux schemes. The physical problem produces smooth variations in flow properties, providing a useful test case. This problem is studied in Ref. [29], with a brief analysis presented here to enhance the current discussion on numerical accuracy. Dimensionless flow properties are considered, with freestream conditions of $\rho_\infty = u_\infty = p_\infty = 1$. The general solution to the problem is given by Eq. 86.

$$\rho\left(x,t\right) = \rho_\infty + Asin\left(\pi\left(x - u_\infty t\right)\right) \tag{86}$$

The amplitude is set to $A = 0.1$, and the grid dimensions are set to $[x_{min}, x_{max}] = [0.0m, 2.0m]$. The initial condition for density is specified according to Eq. 86 at $t = 0$ and is illustrated in Fig. 118. Periodic boundary conditions are specified at the domain edges. The simulations were run for a flow time of $t = 2s$, which results in a final analytic solution that is identical to the initial condition.

Several grid refinements were studied with the number of computational cells ranging from 32 to 256. The time step was set to $\Delta t = 0.002s$, using the $3^{rd}$ order TVD Runge-Kutta time scheme, along with the AUSM+up2 inviscid flux scheme. Comparisons are made between the structured stencil-based numerical methods and the unstructured finite volume methods. The $L_1$, $L_2$, and $L_\infty$ errors are shown in Figs. 119 - 121.

The structured stencil-based approach is identified by the $1^{st}$ Order, the $2^{nd}$ Order MUSCL 2 No Limiter, the $2^{nd}$ Order MUSCL 2 MinMod, and the $3^{rd}$ Order WENO 3. The unstructured stencil-based approach is identified by FVM $1^{st}$ Order, FVM $2^{nd}$ Order No Limiter, and FVM $2^{nd}$ Order Limiter.

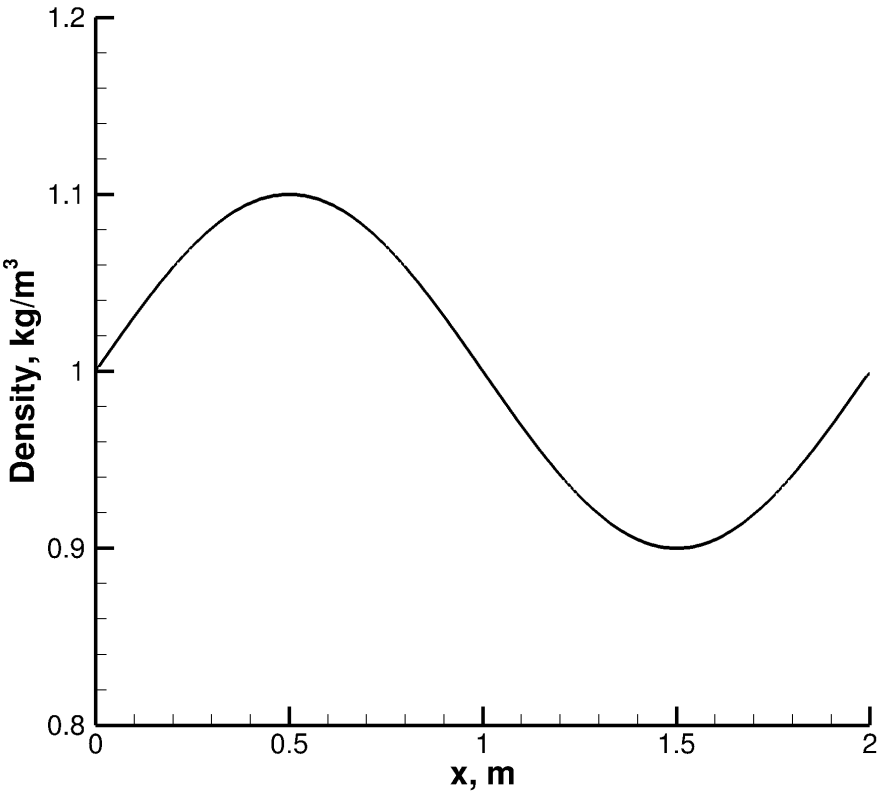



Figure 118: Entropy wave advection: initial condition.

The $L_1$ error plot in Fig. 119, demonstrates the implemented accuracy for all the schemes identified above. It is clear that all schemes using the structured stencil-based approach demonstrate $1^{st}$ order accuracy, despite the intended accuracy ranging from $1^{st}$ to $3^{rd}$ order accuracy. The $1^{st}$ order accurate unstructured stencil-based FVM approach demonstrates $1^{st}$ order accuracy as it should, and agrees with the $1^{st}$ order structured scheme. However, the unstructured FVM approach correctly demonstrates $2^{nd}$ order accuracy. The $L_2$ and $L_\infty$ error plots further demonstrate these results. The implications of these results are significant when considering the use of the structured stencil-based schemes throughout much of this thesis. Discussions on the impact of this are detailed where appropriate throughout the previous sections of this document.

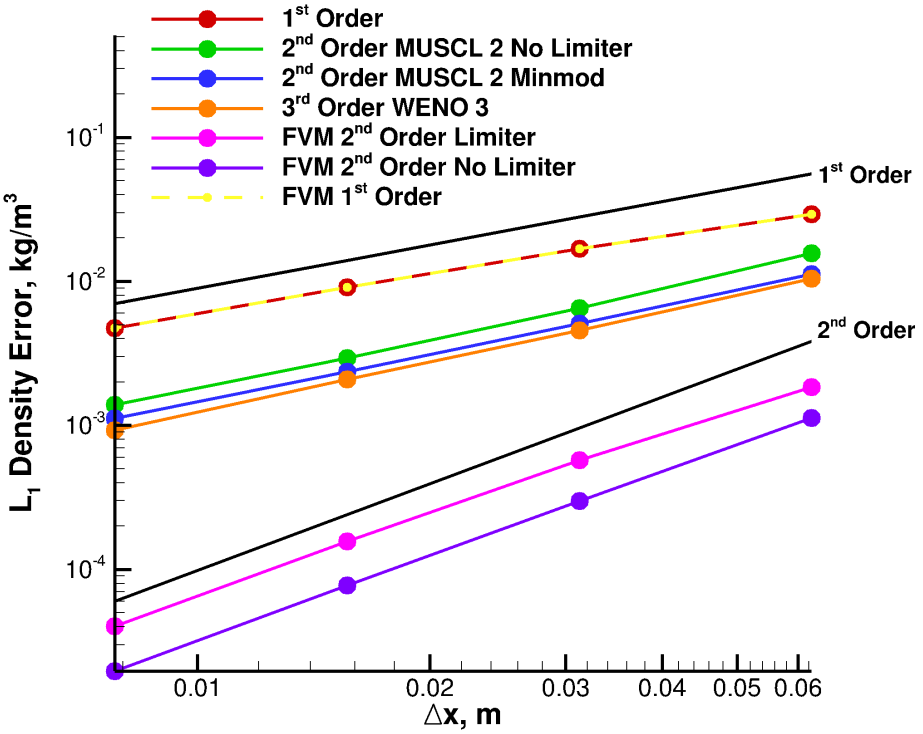



Figure 119: $L_1$ error for entropy wave advection simulations.

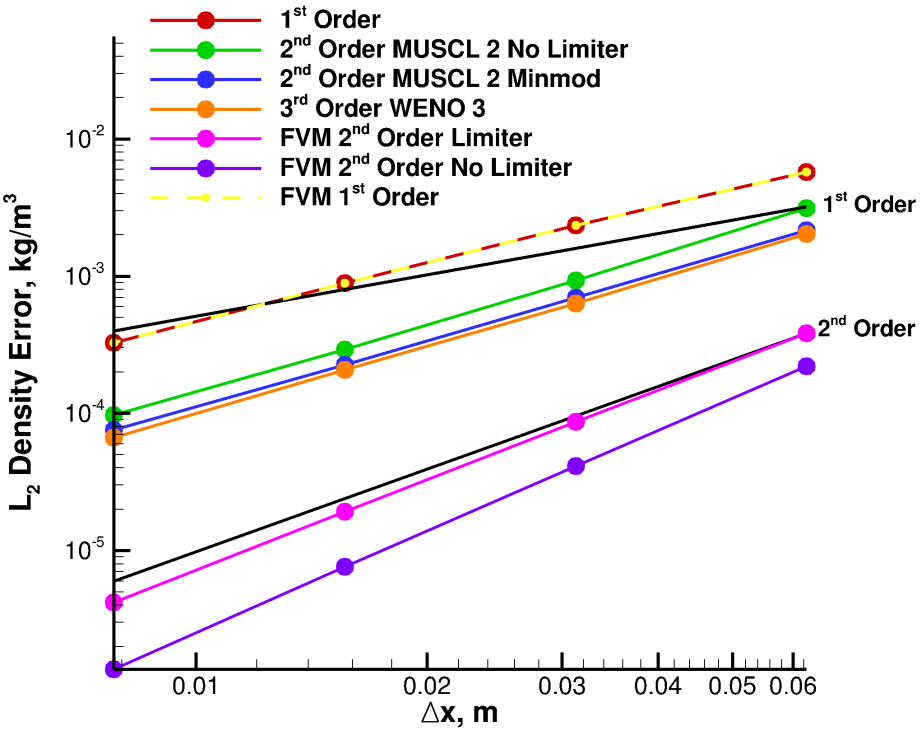



Figure 120: $L_2$ error for entropy wave advection simulations.

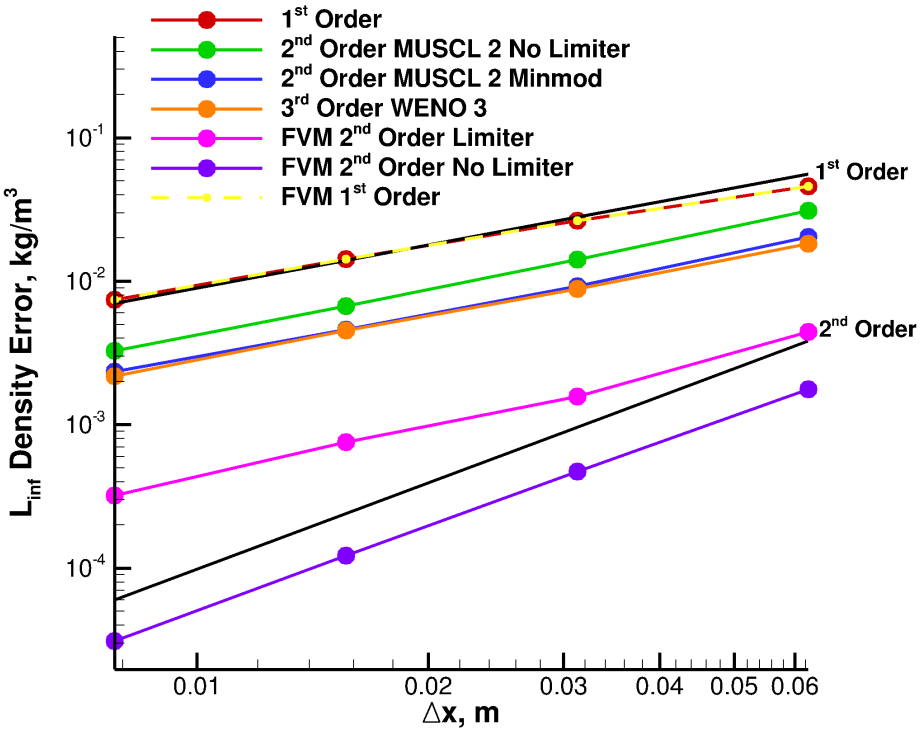



Figure 121: $L_\infty$ error for entropy wave advection simulations.

# REFERENCES


[1] Anderson, J., *Modern Compressible Flow: With Historical Perspective*. McGraw-Hill, 2003.

[2] Anderson, M., Robinson, J., Bush, H., and Fralich, R., "A tension shell structure for application to entry vehicles," Tech. Rep. TN D-2675, NASA Technical Note, 1965.

[3] Anderson, W. and Bonhaus, D., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers & Fluids*, vol. 23, pp. 1–21, 1994.

[4] Anonymous, "Viking lander 'as built' performance capabilities," Tech. Rep. NASA Contract NAS1-9000, Martin Marietta Corp. Report, 1976.

[5] Arslanbekov, R., Kolobov, V., Ruffin, S., Prasad, M., and Zaki, M., "Implementation and evaluation of normal ray refinement technique in adaptive cartesian frameworks," in *AIAA Fluid Dynamics Conference*, AIAA 2012, 2012.

[6] Barnhardt, M., "Development of fluid-structure interaction capability in DPLR," in *AIAA Aerospace Sciences Meeting*, AIAA 2013-0799, 2013.

[7] Batina, J. D., "Unsteady Euler airfoil solutions using unstructured dynamic meshes," *AIAA Journal*, vol. 28, pp. 1381–1388, August 1990.

[8] Bird, G., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. New York, NY: Oxford University Press, 1994.

[9] Blom, F. J., "A monolithical fluid-structure interaction algorithm applied to the piston problem," *Computational Methods in Applied Mechanics and Engineering*, vol. 167, pp. 369–391, December 1998.

[10] Bopp, M., Dement, D., Ruffin, S., and Lee, W., "An improved object-oriented cartesian grid framework implementing three dimensional normal ray refinement," in *AIAA Applied Aerodynamics Conference*, AIAA 2014, 2014.

[11] Braun, R. D. and Manning, R., "Mars exploration entry, descent, and landing challenges," *Journal of Spacecraft and Rockets*, vol. 44, no. 2, pp. 310–323, 2007.

[12] Brehm, C., Barad, M., and Kiris, C., "An immersed boundary method for solving the compressible navier-stokes equations with fluid-structure interaction," in *AIAA Applied Aerodynamics Conference*, AIAA 2016-3265, 2016.

[13] Brehm, C. and Fasel, H., "A non-staggered immersed interface method for solving the incrompessible Navier-Stokes equations," in *AIAA Fluid Dynamics Conference and Exhibit*, AIAA 2010-4433, 2010.

[14] Brehm, C. and Fasel, H., "Immersed interface method for solving the incompressible Navier-Stokes equations with moving boundaries," in *AIAA Aerospace Sciences Meeting*, AIAA 2011-758, 2011.

[15] Brehm, C., Hader, C., and Fasel, H., "Novel immersed boundary/interface method for the compressible Navier-Stokes equations," in *AIAA Aerospace Sciences Meeting*, AIAA 2012-1110, 2012.

[16] Brehm, C., Hader, C., and Fasel, H., "A higher-order immersed boundary method for viscous compressible flows," in *AIAA Theoretical Fluid Mechanics Conference*, AIAA 2014-2093, 2014.

[17] Brehm, C., *Novel Immersed Interface Method for Solving the Incompressible Navier-Stokes Equations*. PhD thesis, Department of Aerospace and Mechanical Engineering, The University of Arizona, 2011.

[18] Bunning, P. G., Jespersen, D. C., Pulliam, T., Klopfer, G. H., and Chan, W. M., "Overflow user's manual," April 2007.

[19] Carczenko, N., "Wind-tunnel investigation of drag and stability of parachutes at supersonic speeds," Tech. Rep. X-991, NASA TM, 1964.

[20] Causin, P., Gerbeau, J., and Nobile, F., "Added-mass effect in the design of partitioned algorithms for fluid-structure problems," *Computational Methods in Applied Mechanical Engineering*, vol. 194, pp. 4506–4527, 2005.

[21] Cirak, F., Deiterding, R., and Mauch, S., "Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin shells subjected to shocks and detonations," *Computers and Structures*, vol. 85, no. 11-14, pp. 1049–1065, 2006.

[22] Clark, I. G., Cruz, J. R., Hughes, M. F., Ware, J. S., Madlangbayan, A., and Braun, R. D., "Aerodynamic and aeroelastic characteristics of a tension cone inflatable aerodynamic decelerator," in *AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA 2000-2967, May 2009.

[23] Clark, I. G., *Aerodynamic Design, Analysis, and Validation of a Supersonic Inflatable Decelerator*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2009.

[24] Coquet, Y., Bordenave, P., Capmas, G., and Espinosa, C., "Improvements in fluid structure interaction simulations of parachutes using LS-DYNA," in *AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA 2011-2590, 2011.

[25] Corp., L. S. T., *LS-DYNA Keyword User's Manual, Version 971*. Oxford University Press, 2007.

[26] Cruz, J. and Lingard, J., "Aerodynamic decelerators for planetary exploration: Past, present, and future," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA 2006-6792, Aug. 2006.

[27] Dadone, A. and Grossman, B., "An immersed body methodology for inviscid flows on cartesian grids," in *AIAA Aerospace Sciences Meeting*, AIAA 2002-1059, 2002.

[28] Deiterding, R., Radovitzky, R., Mauch, S., Noels, L., Cummings, J., and Meiron, D., "A virtual test facility for the efficient simulation of solid materials under high-energy shock-wave loading," *Engineering with Computers*, vol. 22, no. 3-4, pp. 325–347, 2006.

[29] Dement, D. and Ruffin, S., "Higher order cell centered finite volume schemes for unstructured cartesian grids," in *AIAA SciTech*, AIAA, 2018.

[30] Donea, J., Huerta, A., Ponthot, J.-P., and Rodríguez-Ferran, A., *Encyclopedia of Computational Mechanics*, vol. 1. John Wiley & Sons, 2004.

[31] Druguet, M., Candler, G., and Nompelis, I., "Effects of numerics on Navier-Stokes computations of hypersonic double-cone flows," *AIAA Journal*, vol. 43, no. 3, pp. 616–623, 2005.

[32] Farhat, C., Larat, A., Main, A., Avery, P., Wang, K., and Saint-Jalm, C., "An embedded boundary method for viscous fluid/structure interaction problems and application to flexible flapping wings," in *AIAA Fluid Dynamics Conference and Exhibit*, AIAA 2012-2688, 2012.

[33] Farhat, C. and Lesoinne, M., "Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 182, pp. 499–515, February 2000.

[34] Farhat, C., Rallu, A., Wang, K., and Belytschko, T., "Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly non-linear compressible fluid-structure interaction problems," *International Journal for Numerical Methods in Engineering*, vol. 84, pp. 73–107, 2010.

[35] Farhat, C., van der Zee, K., and Geuzaine, P., "Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, pp. 1973–2001, March 2006.

[36] Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)," *Journal of Computational Physics*, vol. 152, no. 2, pp. 457–492, 1999.

[37] Forrer, H. and Berger, M., "Flow simulations on cartesian grids involving complex moving geometries," *International Series of Numerical Mathematics*, vol. 129, pp. 315–324, 1999.

[38] Froehle, B. and Persson, P., "A high-order implicit-explicit fluid-structure interaction method for flapping flight," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2013-2690, 2013.

[39] Gaitonde, A. L. and Fiddes, S. P., "A comparison of a cell-centre method and a cell-vertex method for the solution of two-dimensional unsteady Euler equations on a moving grid," *Journal of Aerospace Engineering*, vol. 209, pp. 203–213, July 1995.

[40] Geuzaine, P., van der Zee, K., and Farhat, C., "Second-order time-accurate loosely-coupled solution algorithms for nonlinear FSI problems," in *European Congress on Computational Methods in Applied Sciences and Engineering*, ECCOMAS 2004, 2004.

[41] Gidzak, V. M., *A method of Simulating Fluid Structure Interactions for Deformable Decelerators*. PhD thesis, University of Minnesota, Minneapolis, MN, 2010.

[42] Gillis, C., "Deployable aerodynamic decelerators for space missions," *Journal of Spacecraft and Rockets*, vol. 6, no. 8, pp. 885–890, 1969.

[43] Gilmanov, A., Acharya, S., and Gilmanov, T., "Flow-structure interaction simulations for ballutes in supersonic flow," in *AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA 2009-2906, 2009.

[44] Giordano, J., Jourdan, G., Burtschell, Y., Medale, M., Zeitoun, D., and Houas, "Shock wave impacts on deforming panel, an application of fluid-structure interaction," *Shock Waves*, vol. 14, pp. 103–110, June 2005.

[45] Gnoffo, P., "An upwind-biased point-implicit relaxation algorithm for viscous, compressible perfect-gas flows," Tech. Rep. 2953, NASA TP, 1990.

[46] Griffin, O., "Vortex-induced vibrations of marine structures in uniform and sheared currents," in *NSF Workshop of Riser Dynamics*, University of Michigan, 1998.

[47] Guennebaud, G., Jacob, B., and others, "Eigen v3, http://eigen.tuxfamily.org," 2010.

[48] Günther, C., Schneiders, L., Meinke, M., Schröder, W., and Hartmann, D., "A Cartesian cut-cell method for sharp moving boundaries," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2011-3387, 2011.

[49] Harris, R. E., "Adaptive Cartesian immersed boundary method for simulation of flow over flexible geometries," *AIAA Journal*, vol. 51, pp. 53–69, January 2013.

[50] Harris, R. E. and Kolobov, V. I., "Adaptive Cartesian immersed boundary method for simulation of flow over flexible lifting surfaces," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2011-3384, 2011.

[51] Houtz, N., "Optimization of inflatable drag devices by isotensoid design," in *AIAA Paper*, 64-437, July 1964.

[52] Jahangirian, A. and Hadidoolabi, M., "An implicit solution of the unsteady Navier-Stokes equations on unstructured moving grids," in *International Congress of the Aeronautical Sciences*, ICAS, 2004.

[53] Jaiman, R., Geubelle, P., Loth, E., and Jiao, X., "Combined interface boundary condition method for unsteady fluid-structure interaction," *Computational Methods in Applied Mechanics and Engineering*, vol. 200, pp. 27–39, January 2011.

[54] John, J., *Gas Dynamics*. Prentice Hall, Englewood, 1984.

[55] Karagiozis, K., Cirak, F., Kamakoti, R., and Pantano, C., "Computational fluid-structure interaction methods for simulation of inflatable aerodynamic decelerators," in *AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA 2009-2968, 2009.

[56] Karagiozis, K., Kamakoti, R., Cirak, F., and Pantano, C., "A computational study of supersonic disk-gap-band parachutes using large eddy simulation coupled to a structural membrane," *Journal of Fluid Mechanics and Structures*, vol. 27, no. 2, pp. 175–192, 2011.

[57] Karimian, S. and Ardakani, M., "Immersed boundary method for the solution of 2D inviscid compressible flow using finite volume approach on moving Cartesian grid," *Journal of Applied Fluid Mechanics*, vol. 4, no. 2, pp. 27–36, 2011.

[58] Kim, K., Kim, C., and Rho, O., "Methods for the accurate computations of hypersonic flows," *Journal of Computational Physics*, vol. 174, pp. 38–80, Jan. 2001.

[59] Kinney, D. J., "Fast methods for fluid structural interactions," in *AIAA Aerospace Sciences Meeting*, AIAA 2009-1104, 2009.

[60] Kinney, D. J., "Prediction of fluid-surface interactions and aerothermal environments and cbaero," in *AIAA Aerospace Sciences Meeting*, AIAA 2011-1186, 2011.

[61] Kiris, C., Barad, M., Housman, J., Sozer, E., Brehm, C., and Moini-Yekta, S., "The LAVA computational fluid dynamics solver," in *AIAA SciTech*, AIAA 2014, 2014.

[62] Kramer, R., Cirak, F., and Pantano, C., "Fluid-structure interaction simulations of a tension-cone inflatable aerodynamic decelerator," *AIAA Journal*, vol. 51, pp. 1640–1656, July 2013.

[63] Lakshminarayan, V. and Farhat, C., "An ALE Eulerian formulation of embedded boundary methods for turbulent fluid-structure interaction problems," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2013-2441, 2013.

[64] Lakshminarayan, V., Farhat, C., and Main, A., "An embedded boundary framework for compressible turbulent flow and fluid-structure computations on structured and unstructured grids," *International Journal for Numerical Methods in Fluids*, vol. 76, no. 6, pp. 366–395, 2014.

[65] Landon, R., "Compendium of unsteady aerodynamic measurements," Tech. Rep. 702, AGARD Report, Aug. 1982.

[66] LeBeau, G. and Lumpkin, F., "Application highlights of the DSMC analysis code (DAC) software for simulating rarefied flows," *Computer Methods in Applies Mechanics and Engineering*, vol. 191, no. 6-7, pp. 595–609, 2001.

[67] Lee, J. D., *Development of an Efficient Viscous Approach in a Cartesian Grid Framework and Application to Rotor-Fuselage Interaction.* PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2006.

[68] Lee, J. W., *Parallelized Cartesian Grid Methodology for Non-equilibrium Hypersonic Flow Analysis of Ballutes*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2007.

[69] Lee, J. and Ruffin, S., "Parallel computation of solution adaptive Cartesian grids with SFC," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2007-4088, Jan. 2007.

[70] Lesoinne, M. and Farhat, C., "Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems," *AIAA Journal*, vol. 36, pp. 1754–1757, 2006.

[71] Lin, C. Q. and Pahlke, K., "Numerical solution of Euler equations for aerofoils in arbitrary unsteady motion," *Aeronautical Journal*, vol. 98, pp. 207–214, June–July 1994.

[72] Liou, M.-S., "The root cause of the numerical overheating problem," in *AIAA Computational Fluid Dynamics Conference*, AIAA, 2017.

[73] Marshall, D. and Ruffin, S., "An embedded boundary condition grid scheme for viscous flows using a new viscous wall boundary condition treatment," in *AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2004-0581, Jan. 2004.

[74] Marshall, D. and Ruffin, S., "A new inviscid wall boundary condition treatment for embedded boundary Cartesian grid schemes," in *AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2004-0583, Jan. 2004.

[75] Marshall, D. D., *Extending the Functionalities of Cartesian Grid Solvers: Viscous Effects Modeling and MPI Parallelization*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2002.

[76] Mayhue, R. and Bobbitt, P., "Drag characteristics of a disk-gap-band parachute with a nominal diameter of 1.65 meters at Mach numbers from 2.0 to 3.0," Tech. Rep. D-6894, NASA TN, 1972.

[77] Maynard, J., "Aerodynamic characteristics of parachutes at Mach numbers from 1.6 to 3," Tech. Rep. D-752, NASA TN, 1961.

[78] Michler, C., Hulshoff, S. J., van Brummelen, E. H., and de Borst, R., "A monolithic approach to fluid-structure interaction," *Computers and Fluids*, vol. 33, pp. 839–848, June–July 2004.

[79] Mirsajedi, S., Karimian, S., and Mani, M., "A multizone moving mesh algorithm for simulation of flow around a rigid body with arbitrary motion," *ASME Journal of Fluids Engineering Engineering*, vol. 128, no. 2, pp. 297–304, 2006.

[80] Mittal, R. and Iaccarino, G., "Immersed boundary methods," *Annual Review of Fluid Mechanics*, vol. 37, pp. 239–261, 2005.

[81] Muralidaran, B. and Menon, S., "Simulations of unsteady shocks and detonation interactions with structures," in *AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, AIAA 2013-3655, 2013.

[82] Murman, S. and Aftosmis, M., "Dynamic analysis of atmospheric-entry probes and capsules," in *AIAA Aerospace Sciences Meeting*, AIAA 2007-74, 2007.

[83] Murman, S. M., Aftosmis, M. J., and Berger, M. J., "Implicit approaches for moving boundaries in a 3-D Cartesian method," in *AIAA Aerospace Sciences Meeting*, AIAA 2003-1119, Jan. 2003.

[84] Murman, S. M., Aftosmis, M. J., and Berger, M. J., "Simulations of 6-DOF motion with a Cartesian method," in *AIAA Aerospace Sciences Meeting*, AIAA 2003-1246, Jan. 2003.

[85] Newman, D. and Karniadakis, G., "Simulations of flow over a flexible cable: A comparison of forced and flow-induced vibration," *Journal of Fluids and Structures*, vol. 10, pp. 439–453, July 1996.

[86] Noh, W. F., "Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux," *Journal of Computational Physics*, vol. 72, pp. 78–120, Oct. 1978.

[87] Nompelis, I., Drayna, T., and Candler, G., "A parallel unstructured implicit solver for hypersonic reacting flow simulations," in *AIAA Paper*, 2010-2830, June 2005.

[88] Nompelis, I., Drayna, T., and Candler, G., "A parallel unstructured implicit solver for hypersonic reacting flow simulation," in *AIAA Fluid Dynamics Conference*, AIAA 2005-4867, 2005.

[89] Piperno, S., "Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcyling for 2D inviscid aeroelastic simulations," *International Journal for Numerical Methods in Fluids*, vol. 25, pp. 1207–1226, November 1997.

[90] Piperno, S., *Simulation numerique de phenomenes d'interaction fluide-structure*. PhD thesis, Mathematics, Ecole des Ponts ParisTech, 1995.

[91] Potvin, J., Bergeron, K., Brown, G., Charles, R., Desabrais, K., Johari, H., V., K., McQuilling, M., Morris, A., Noetscher, G., and Tutt, B., "The road ahead: A white paper on the development, testing and use of advanced numerical modeling for aerodynamic decelerator systems design and analysis," in *AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA 2011-2501, 2011.

[92] Richard, J. and Nicoud, F., "Effect of the fluid structure interaction on the aeroacoustic instabilities of solid rocket motors," in *AIAA/CEAS Aeroacoustics Conference*, AIAA 2011-2816, 2011.

[93] Rohrschneider, R. R., *Variable-Fidelity Hypersonic Aeroelastic Analysis of Thin-Film Ballutes for Aerocapture*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2007.

[94] Ruffin, S., Zaki, M., Kumar, V., and Sekhar, S., "A normal ray refinement technique for cartesian-grid based navier-stokes solver," in *AIAA Aerospace Sciences Meeting*, AIAA 2011, 2011.

[95] Ruffin, S., Zaki, M., and Sekhar, S., "A normal ray refinement technique for cartesian-grid based navier-stokes solver," *International Journal of Computational Fluid Dynamics*, vol. 26, July 2012.

[96] Samareh, J. A., "Discrete data transfer technique for fluid-structure interaction," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2007-4309, June 2007.

[97] Schneiders, L., Meinke, M., and Schröder, W., "A robust cut-cell method for fluid-structure interaction on adaptive meshes," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2013-2716, 2013.

[98] Schoenenberger, M., Yates, L., and Hathaway, W., "Dynamic stability testing of the mars science laboratory entry capsule," in *AIAA Thermophysics Conference*, AIAA 2009-3917, June 2009.

[99] Sekhar, S. K., *Viscous Hypersonic Flow Physics Predictions Using Unstructured Cartesian Grid Techniques*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2012.

[100] Sengupta, A., Wernet, M., Roeder, J., Kelsch, R., Witkowski, A., and Jones, T., "Supersonic testing of 0.8 m disk gap band parachutes in the wake of a 70 deg sphere cone entry vehicle," in *20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2009-2900, 2009.

[101] Smith, B., Tanner, C., Mahzari, M., Clark, I., and Braun, R., "A historical review of inflatable aerodynamic decelerator technology development," in *IEEE Aerospace Conference*, IEEE 2010-1276, Aug. 2010.

[102] Smith, M. J., Hodges, D. H., and Cesnik, C. E. S., "Evaluation of computational algorithms suitable for fluid-structure interactions," *Journal of Aircraft*, vol. 37, pp. 282–294, March-April 2000.

[103] Smith, M., Cesnik, C. E. S., Hodges, D., and Moran, K., "An evaluation of computational algorithms to interface between CFD and CSD," in *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit*, AIAA 1996, 1996.

[104] Sterenborg, J., van Zuijlen, A., and Bijil, H., "Solution based mesh adaption applied to fluid structure interaction computations," in *AIAA Aerospace Sciences Meeting*, AIAA 2009-581, 2009.

[105] Stern, E., Gidzak, V., and Candler, G., "Estimation of dynamic stability coefficients for aerodynamic decelerators using CFD," in *AIAA Applied Aerodynamics Conference*, AIAA 2012-3225, June. 2012.

[106] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*. Taylor & Francis, 1997.

[107] Tanner, C. L., *Aeroelastic Analysis and Testing of Supersonic Inflatable Aerodynamic Decelerators*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2012.

[108] Tu, S., *Development of a Solution Adaptive Cartesian-Grid Solver for 2-D Thermochemical Nonequilibrium Flows*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2001.

[109] Tutt, B., "Fluid structure interaction parachute benchmark models in LS-DYNA," in *AIAA Aerodynamic Decelerator Systems Technology Conferences*, AIAA 2013-1384, 2013.

[110] Tutt, B., Roland, S., Charles, R., and Noetscher, G., "Finite mass simulation techniques in LS-DYNA," in *AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA 2011-2592, 2011.

[111] van Brummelen, E., "Added mass effects of compressible and incompressible flows in fluid-structure interaction," *Journal of Applied Mechanics*, vol. 76, pp. 1–9, March 2009.

[112] van Zuijlen, A. H. and Bijl, H., "High order time integration for fluid-structure interaction on moving meshes," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2005-5247, 2005.

[113] Wang, K., Grétarsson, J., Main, A., and Farhat, C., "Computational algorithms for tracking dynamic fluid-structure interfaces in embedded/immersed boundary methods," in *AIAA Computational Fluid Dynamics Conference*, AIAA 2011-3385, 2011.

[114] Wang, K., Grétarsson, J., Main, A., and Farhat, C., "Computational algorithms for tracking dynamic fluid-structure interfaces in embedded boundary methods," *International Journal for Numerical Methods in Fluids*, vol. 70, pp. 515–535, April 2012.

[115] Wang, K., Rallu, A., Gerbeau, J.-F., and Farhat, C., "Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid-structure interaction problems," *International Journal for Numerical Methods in Fluids*, vol. 67, pp. 1175–1206, April 2011.

[116] Wright, M., Candler, G., and Bose, D., "Data-Parallel Line Relaxation method for the Navier-Stokes equations," *AIAA Journal*, vol. 36, pp. 1603–1609, 1998.

[117] Zaki, M. and Ruffin, S., "Conservation and grid adaption enhancements to a normal ray refinement technique for cartesian-grid based navier-stokes solvers," in *AIAA Aerospace Sciences Meeting*, AIAA 2012, 2012.

[118] Zhou, C., So, R., and Lam, K., "Vortex-induced vibrations of an elastic circular cylinder," *Journal of Fluids and Structures*, vol. 13, pp. 165–189, February 1999.